



**RF EMITTER TRACKING AND INTENT
ASSESSMENT**

THESIS

Benjamin J. Kuhar, Captain, USAF

AFIT-ENG-13-M-29

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

**DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED**

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-13-M-29

RF EMITTER TRACKING AND INTENT
ASSESSMENT

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Benjamin J. Kuhar, B.S. Cp.E.

Captain, USAF

March 2013

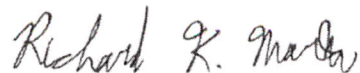
DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT-ENG-13-M-29

RF EMITTER TRACKING AND INTENT
ASSESSMENT

Benjamin J. Kuhar, B.S. Cp.E.
Captain, USAF

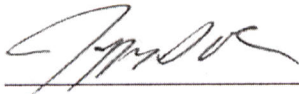
Approved:



Dr. R.K. Martin, PhD (Chairman)

8 Mar 2013

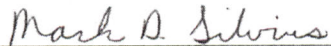
Date



Lt Col J.D. Clark, PhD (Member)

8 Mar 2013

Date



Maj M.D. Silvius, PhD (Member)

8 Mar 2013

Date

Abstract

Current research in employing pattern recognition techniques in a wireless sensor network (WSN) to detect anomalous or suspicious behavior is limited. The purpose of this research was to determine the feasibility of an accurate tracking and intent assessment system of unknown or foreign radio frequency (RF) emitters in close proximity to and within military installations as a method for physical security.

22 position tracks were collected using a hand-held Global Positioning System (GPS) unit and a training data set from five different features was generated for each position track. Each collected position track was individually classified as suspicious or non-suspicious by the leave-one-out-cross-validation (LOOCV) method using four different classification methods. The four classification methods used in this research were the linear discriminant function (LDF), the diagonal linear discriminant function (DLDF), the quadratic discriminant function (QDF) and the Mahalanobis distance method. The accuracies and false positive/negative error rates of the four classification methods were compared for different assessment system configurations. Additionally, best fit receiver operating characteristic (ROC) curves were generated for each classification method and discussed.

The QDF classification method out-performed the other three classification methods. This classification method achieved an accuracy of 95% when it classified the 22 position tracks one at a time. The lowest false positive and false negative rates were 10% and 0%, respectively. The prior probabilities for the non-suspicious and suspicious classes were both set to 50% class for this configuration.

Acknowledgments

I would first like to thank Dr. Martin for your steady hand that guided this research effort. You consistently maintained the bigger picture and always communicated that direction to me. Thank you for spending countless hours looking at my results and drafts. I earnestly wish you the best of luck with your future endeavors at AFIT or wherever they may be, educating bright young minds. Thank you Colonel Clark and Major Silviu for asking the tough questions and opening my eyes to things I had not thought of. I am also thankful for the time spent with classmates and for the help received with the coursework and L^AT_EX questions I had. Finally, I am eternally grateful for the support of my friends and family members over the past year and a half. Your loyalty and encouragement will not be forgotten.

Thank you God for bringing me through AFIT. You are worthy of all praise and have made everything beautiful in its time.

Where shall I go from your Spirit? Or where shall I flee from your presence? If I ascend to heaven, you are there! If I make my bed in Sheol, you are there! If I take the wings of the morning and dwell in the uttermost parts of the sea, even there your hand shall lead me, and your right hand shall hold me. – *Psalm 139:7-10*

Benjamin J. Kuhar

Table of Contents

| | Page |
|--|-------|
| Abstract | iv |
| Acknowledgments | v |
| Table of Contents | vi |
| List of Figures | ix |
| List of Tables | xii |
| List of Acronyms | xiii |
| I. Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Problem Statement | 3 |
| 1.3 Scope and Application | 3 |
| 1.4 Research Objectives | 4 |
| 1.5 Assumptions | 4 |
| 1.6 Equipment Used | 4 |
| 1.7 Motivation | 5 |
| 1.8 Thesis Organization | 5 |
| II. Background | 6 |
| 2.1 Source Localization | 6 |
| 2.1.1 Time of Arrival | 7 |
| 2.1.2 Time Distance of Arrival | 8 |
| 2.1.3 Angle of Arrival | 10 |
| 2.1.4 Received Signal Strength | 11 |
| 2.2 Pattern Recognition | 17 |
| 2.2.1 Data Collection | 17 |
| 2.2.2 Segmentation | 17 |
| 2.2.3 Feature Generation | 17 |
| 2.2.4 Classification | 18 |
| 2.2.4.1 LDA | 18 |

| | Page |
|---|------|
| 2.2.4.2 LDF | 19 |
| 2.2.4.3 DLDF | 21 |
| 2.2.4.4 QDF | 21 |
| 2.2.4.5 Mahalanobis Method | 22 |
| 2.2.4.6 Classifier Terminology | 23 |
| 2.2.4.7 Reporting Methods | 24 |
| 2.3 Pattern Recognition Applied to WSNs | 25 |
| 2.4 Pattern Recognition Applied to Cyber Security | 27 |
| 2.5 Chapter Summary | 29 |
| III. Research Methodology | 31 |
| 3.1 Magellan® GPS Unit Operation | 32 |
| 3.2 Position Track Processing | 34 |
| 3.2.1 Importing GPS Data into MATLAB® | 34 |
| 3.2.2 Lat-Long to x-y Coordinate Conversion | 34 |
| 3.3 Feature Generation | 36 |
| 3.3.1 Development of Grid | 36 |
| 3.3.2 Dwell | 37 |
| 3.3.3 Repetition | 40 |
| 3.3.4 Development of Landmark Distance Maps | 42 |
| 3.3.5 Deviation from Roads and Parking Lots | 44 |
| 3.3.6 Proximity to High-valued Building | 47 |
| 3.3.7 Proximity to Water Tower | 49 |
| 3.4 Classification of Position Tracks | 52 |
| 3.4.1 Pre-classification Data Processing | 52 |
| 3.4.2 Classification in MATLAB® | 54 |
| 3.5 Chapter Summary | 55 |
| IV. Results and Analysis | 56 |
| 4.1 Feature Generation Results | 56 |
| 4.2 Classifier Results | 62 |
| 4.2.1 Grid Resolution Sweep Results | 63 |
| 4.2.2 Prior Probability Sweep Results | 71 |
| 4.2.3 ROC Curve Analysis | 86 |
| 4.3 Chapter Summary | 95 |

| | Page |
|---|------|
| V. Summary, Conclusions and Future Work | 96 |
| 5.1 Summary | 96 |
| 5.2 Conclusions | 97 |
| 5.3 Future Work | 98 |
| 5.3.1 Implementing Accurate Real-time Geolocation with TelosB Motes . | 98 |
| 5.3.2 Tracking Multiple Emitters Simultaneously in a WSN | 98 |
| 5.3.3 Increasing the Size of the WSN | 99 |
| 5.3.4 Increasing the Size of the Position Track Database | 99 |
| 5.3.5 Improving Feature Generation | 99 |
| 5.3.6 Using Different Classification Methods | 101 |
| 5.4 Chapter Summary | 101 |
| Appendix: Detailed Classifier Results | 102 |
| Bibliography | 115 |

List of Figures

| Figure | Page |
|--|------|
| 1.1 A Typical WSN | 2 |
| 2.1 Tri-lateration TOA | 8 |
| 2.2 Hyperbolic-lateration TDOA | 9 |
| 2.3 Triangulation using Two Receivers | 11 |
| 2.4 Triangulation using Three Receivers | 11 |
| 2.5 Simulated RSS Geolocation Scenarios | 13 |
| 2.6 Emitter Estimates for Sensor Configuration One | 14 |
| 2.7 Emitter Estimates for Sensor Configuration Two | 15 |
| 2.8 Emitter Estimates for Sensor Configuration Three | 16 |
| 2.9 Ellipse-shaped data representation | 22 |
| 2.10 Anomaly Detection in a WSN | 26 |
| 2.11 PNN Architecture | 28 |
| 3.1 Magellan [®] Mobile Mapper GPS Unit | 32 |
| 3.2 Position Track Collection Area | 33 |
| 3.3 Example Position Track | 35 |
| 3.4 Variable Resolution Grid | 37 |
| 3.5 Position Track with Low Dwell Time | 39 |
| 3.6 Position Track with High Dwell Time | 40 |
| 3.7 Position Track with Low Repetition | 41 |
| 3.8 Position Track with High Repetition | 42 |
| 3.9 Deviation from Roads and Parking Lots Distance Map | 46 |
| 3.10 Proximity to High-valued Building Distance Map | 49 |

| Figure | Page |
|---|------|
| 3.11 Proximity to Water Tower Distance Map | 51 |
| 3.12 Example of the Sample, Training Data and Group Matrices | 54 |
| 4.1 Dwell Time vs. Repetition | 57 |
| 4.2 Deviation from Roads and Parking Lots vs. Repetition | 59 |
| 4.3 Proximity to Water Tower vs. Dwell Time | 60 |
| 4.4 Proximity to High-Valued Building vs. Repetition | 62 |
| 4.5 Overall Classifier Accuracy for Grid Resolution Sweep | 64 |
| 4.6 P_M Performance for Grid Resolution Sweep | 67 |
| 4.7 P_F Classifier Performance for Grid Resolution Sweep | 69 |
| 4.8 Classifier Accuracy for $P[H_1]$ Sweep: $w_{cell} = 270$ feet. | 73 |
| 4.9 P_M Performance for $P[H_1]$ Sweep: $w_{cell} = 270$ feet. | 75 |
| 4.10 P_F Performance for $P[H_1]$ Sweep: $w_{cell} = 270$ feet. | 77 |
| 4.11 Classifier Accuracy for $P[H_1]$ Sweep: $w_{cell} = 918$ feet. | 80 |
| 4.12 P_M Performance for $P[H_1]$ Sweep: $w_{cell} = 918$ feet. | 82 |
| 4.13 P_F Performance for $P[H_1]$ Sweep: $w_{cell} = 918$ feet. | 84 |
| 4.14 P_D vs. P_F Data Points and Best Fit ROC Curves: $w_{cell} = 270$ feet | 87 |
| 4.15 Best Fit ROC Curves: $w_{cell} = 270$ feet | 89 |
| 4.16 P_D vs. P_F Data Points and Best Fit ROC Curves: $w_{cell} = 918$ feet | 91 |
| 4.17 Best Fit ROC Curves: $w_{cell} = 918$ feet | 93 |
| A.1 Track 1 on Overhead Imagery | 103 |
| A.2 Track 2 on Overhead Imagery | 104 |
| A.3 Track 3 on Overhead Imagery | 104 |
| A.4 Track 4 on Overhead Imagery | 105 |
| A.5 Track 5 on Overhead Imagery | 105 |

| Figure | Page |
|---|------|
| A.6 Track 6 on Overhead Imagery | 106 |
| A.7 Track 7 on Overhead Imagery | 106 |
| A.8 Track 8 on Overhead Imagery | 107 |
| A.9 Track 9 on Overhead Imagery | 107 |
| A.10 Track 10 on Overhead Imagery | 108 |
| A.11 Track 11 on Overhead Imagery | 108 |
| A.12 Track 12 on Overhead Imagery | 109 |
| A.13 Track 13 on Overhead Imagery | 109 |
| A.14 Track 14 on Overhead Imagery | 110 |
| A.15 Track 15 on Overhead Imagery | 110 |
| A.16 Track 16 on Overhead Imagery | 111 |
| A.17 Track 17 on Overhead Imagery | 111 |
| A.18 Track 18 on Overhead Imagery | 112 |
| A.19 Track 19 on Overhead Imagery | 112 |
| A.20 Track 20 on Overhead Imagery | 113 |
| A.21 Track 21 on Overhead Imagery | 113 |
| A.22 Track 22 on Overhead Imagery | 114 |

List of Tables

| Table | Page |
|---|------|
| 4.1 Confusion Matrix for LDF and DLDF Methods: $w_{cell} = 270$ feet | 65 |
| 4.2 Confusion Matrix for QDF and Mahalanobis Methods: $w_{cell} = 270$ feet | 66 |
| 4.3 Classifier Statistics for Grid Resolution Sweep | 70 |
| 4.4 Classifier Statistics for $P[H_1]$ Sweep: $w_{cell} = 270$ feet. | 78 |
| 4.5 Classifier Statistics for $P[H_1]$ Sweep: $w_{cell} = 918$ feet. | 85 |
| A.1 Classifier Database | 102 |
| A.2 Classifier Database | 103 |

List of Acronyms

| Acronym | Definition | |
|----------|------------------------------------|----|
| DoD | Department of Defense | 1 |
| RF | Radio Frequency | 1 |
| US | United States | 1 |
| LBS | Location Based Services | 1 |
| GPS | Global Positioning System | 1 |
| WSN | Wireless Sensor Network | 2 |
| Lat-Long | Latitude-Longitude | 3 |
| WPAFB | Wright Patterson Air Force Base | 3 |
| MATLAB | Matrix Laboratory® | 3 |
| LOOCV | Leave-One-Out-Cross-Validation | 4 |
| RSS | Received Signal Strength | 4 |
| AFIT | Air Force Institute of Technology | 5 |
| TOA | Time of Arrival | 6 |
| TDOA | Time Distance of Arrival | 6 |
| AOA | Angle of Arrival | 6 |
| DOA | Direction of Arrival | 6 |
| LOB | Line-of-Bearing | 10 |
| RSSI | Received Signal Strength Indicator | 11 |
| dB | decibel | 12 |
| MLE | Maximum Likelihood Estimator | 12 |
| CI | Confidence Interval | 13 |
| CRLB | Cramér-Rao Lower Bound | 13 |
| PDF | Probability Density Function | 13 |

| Acronym | Definition |
|---------|--|
| AI | Artificial Intelligence 17 |
| LDF | Linear Discriminant Function 18 |
| DLDF | Diagonal Linear Discriminant Function 18 |
| QDF | Quadratic Discriminant Function 18 |
| MDA | Multiple Discriminant Analysis 18 |
| ROC | Receiver Operating Characteristic 24 |
| NN | Neural Network 27 |
| DDoS | Distributed Denial of Service 27 |
| GN | Graph Neuron 27 |
| PNN | Probabilistic Neural Network 28 |
| USB | Universal Serial Bus 34 |
| NMEA | National Marine Electronics Association 34 |
| MAC | Media Access Control 98 |
| RBF | Radial Basis Function 101 |

RF EMITTER TRACKING AND INTENT ASSESSMENT

I. Introduction

This chapter introduces this research and provides a brief background on the increasing prevalence of geolocation technologies to locate emitters. It includes the problem statement, research objectives, limitations, equipment required, and a section communicating the importance of this research to the Department of Defense (DoD).

1.1 Background

Research in the area of locating and tracking Radio Frequency (RF) signal-emitting sources has been conducted for the last six decades [1]. The process of determining an unknown position of an emitter is called source localization or geolocation. The capability to geolocate an emitting object is currently a critical requirement necessary to perform certain missions of the United States (US) Military. The ability to determine the precise location of US military personnel on the earth and navigate their movement requires geolocation. Additionally, millions of dollars are invested each year into the continuing research and advancement of Location Based Services (LBS) in the commercial telecommunications sector. In 2010 there were 6,000 location-based applications for the iPhone, 900 for the Android and 300 for the Blackberry [2].

An example of a commercial application that uses geolocation is navigation. Currently, geolocation can be accomplished by both terrestrial-based (ground) and space-based systems. Global Positioning System (GPS) is an example of a system used to

geolocate objects in motion both on the surface of the earth and airborne. Both automobiles and airplanes benefit from geolocation-aided navigation [3].

Geolocation is also employed in the military for locating and tracking both hostile and foreign unknown targets. Wireless Sensor Networks (WSNs) are one method for accomplishing this. WSNs can be used to determine the position or location of a foreign emitter within the network. They are comprised of light-weight devices referred to as sensor nodes. These multi-function nodes are designed to sense their environment, process data and communicate with each other using radio waves [4]. Figure 1.1 is a diagram of a typical WSN. The base station is the central communications focal point, receiving and transmitting pertinent information from and to the sensor nodes.

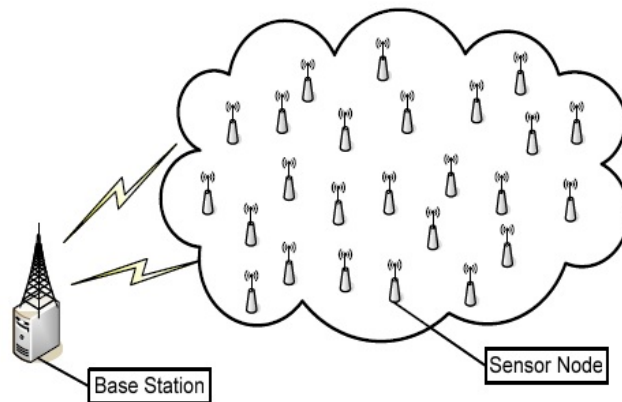


Figure 1.1: A Typical WSN, used with permission [5].

With the addition of a geolocation method to record estimated foreign emitter positions at certain time samples within the network, a real-time emitter tracking system can be created. The tracking data produced by this system can be used to generate feature data which is employed by a pattern recognition mechanism called a classifier.

The classifier will use training data to determine if the emitter exhibits non-suspicious or suspicious (anomalous) behavior.

The system described in the previous paragraph is an example of a tracking and intent assessment system. A human can be placed in the assessment loop and is cued based on the classifier's decision that the position track is suspicious. Chapter II will explain in detail different methods employed for geolocation. That chapter will also explain how WSNs can also be used to detect anomalous behavior in the computer network or cyber domain in addition to the physical domain that was discussed in this section.

1.2 Problem Statement

Current research in the area of employing pattern recognition techniques for detecting suspicious or anomalous behavior within a sensor network is limited. The purpose of this research was to determine the feasibility of an accurate intent assessment system of unknown or foreign emitters for military installations as a method for physical security. Another goal of this research effort was to create geolocated position tracks in real-time using wireless sensors and an estimation algorithm. These position tracks were to be employed as training data for the intent assessment system. However, due to hardware issues with the sensors, a hand-held GPS unit was instead used create the position tracks.

1.3 Scope and Application

Latitude-Longitude (Lat-Long) position tracks were collected using a GPS unit inside a pre-determined area within Wright Patterson Air Force Base (WPAFB), Area B. This data collection area was 2,095 feet by 2,095 feet or 4,389,025 feet² (about 100 acres). Google Maps[®] was used to create an overhead satellite image for the data collect area. In addition, this application was also used to create a reference point on the data collect area to convert the position track data in Matrix Laboratory[®] (MATLAB) from Lat-Long coordinates to x-y coordinate data. Feature data was generated for each position track in the database which

was fed into a classifier. The classifier classified each position track as non-suspicious or suspicious using the Leave-One-Out-Cross-Validation (LOOCV) method.

A considerable amount of time and effort was spent accomplishing certain tasks for this research. Time was required to create position tracks with the GPS unit. Additionally, an interface was created with the aid of Google Maps® to precisely plot the collected position tracks on an overhead image and to convert image pixel distances into an acceptable unit of measure (feet). A third task undertaken was to store all the position tracks in a single matrix in MATLAB® to be efficiently passed into the feature generation algorithms. These tasks are described in detail in chapter III.

1.4 Research Objectives

The main objective of this research was to determine if position tracks collected within a WSN could be accurately classified as non-suspicious or suspicious. Additionally, a secondary goal was to determine if accurate position tracks could be created from real-time geolocated Received Signal Strength (RSS) data of an emitter in a WSN.

1.5 Assumptions

Certain limitations and assumptions were established for this research effort. First, the emitter that was tracked inside the WSN could exist in a state of motion. Secondly, the intent assessment system designed in this research could only locate and track one emitter in real-time.

1.6 Equipment Used

A Magellan® Mobile Mapper GPS unit was used to collect the position tracks for this research. Additionally, MATLAB® was used to process all the data pertaining to this research and to generate the results. All of the MATLAB® code developed for this research was original except for three functions. The first function imported data from the hand-held GPS unit into MATLAB®. The second function converted Lat-Long coordinates to x-y

coordinate data. A third function created landmark distance maps using landmark bitmap images. These functions were written by a prior Air Force Institute of Technology (AFIT) student.

1.7 Motivation

Protecting the integrity of military installations is paramount to the ongoing operations of the US Military. Breakthroughs in this research area would inevitably have a positive impact on the security of both domestic and deployed military bases alike. The concept of an automated intent assessment system poses multiple advantages to the military.

This system would reduce the required security manning associated with perimeter surveillance by determining the intent of personnel approaching military installations [6]. By reducing the manning of security personnel, the operating costs required to keep the installation at a normal operating level would decrease. The security personnel displaced from the task of monitoring perimeter activity could be employed in some other fashion.

Additionally, the physical security infrastructure of the installation would improve due to a precise, real-time tracking and intent assessment system installed to monitor all RF emitter activity within close proximity to and inside the installation. Suspicious emitter activity would be flagged so that security personnel could take the proper action.

1.8 Thesis Organization

This thesis is organized into five chapters. Chapter II presents a background on geolocation and pattern recognition methods in the context of this research. Chapter III explains the processes and methodologies used in this research. Chapter IV presents the analytical results of this research. Finally, chapter V summarizes this thesis, provides a conclusion and discusses areas of possible future research.

II. Background

This chapter introduces the theory and principles relevant to RF emitter tracking and intent assessments. The fundamental principles of source localization and pattern recognition are discussed as well as current research thrusts in the areas of anomaly detection and pattern recognition in a WSN to aid in computer network security.

With the demand increase for heightened security both in the physical and cyber domains, an emphasis has been placed on advancing research in the area of developing sophisticated algorithms that detect and prevent intrusions [4]. Pattern recognition techniques that detect anomalous events in WSNs are viable solutions for meeting this demand. No existing literature was found on previous research attempts to determine the feasibility of RF emitter tracking and intent assessment systems designed specifically to improve a military installation's security.

2.1 Source Localization

Geolocation or source localization is the process of estimating the position or location of an RF transmitter. There are multiple ways to perform geolocation. Four common geolocation methods are Time of Arrival (TOA), Time Distance of Arrival (TDOA), Angle of Arrival (AOA) and RSS [7].

A conference paper introduced an RF geolocation and tracking system for the US Navy. The paper communicated that the Navy requires accurate detecting, locating and tracking of mobile RF emitters. The system design used Direction of Arrival (DOA) (similar to AOA) as the geolocation method and proposed a Kalman filter for the tracking mechanism [8]. This system did not perform intent assessments of the emitters.

2.1.1 Time of Arrival

TOA is a localization technique that measures the absolute time at which a transmitted signal first arrives at a receiver. If two receivers are in operation, TOA data will determine the estimated transmitter position to be one of two equally probable points. If three receivers are employed, a single, precise position estimate results through a process called tri-lateration. This process involves determining the emitter position by using the intersection of ranging circles. Figure 2.1 illustrates the tri-lateration process. A, B and C are the sensors, X is the emitter, and $r_{A,B,C}$ are the radial distances from each sensor to X. It is possible in certain cases that tri-lateration will produce more than one estimate for the emitter. Multi-lateration employs at least four sensors and improves the accuracy of the localization process [1].

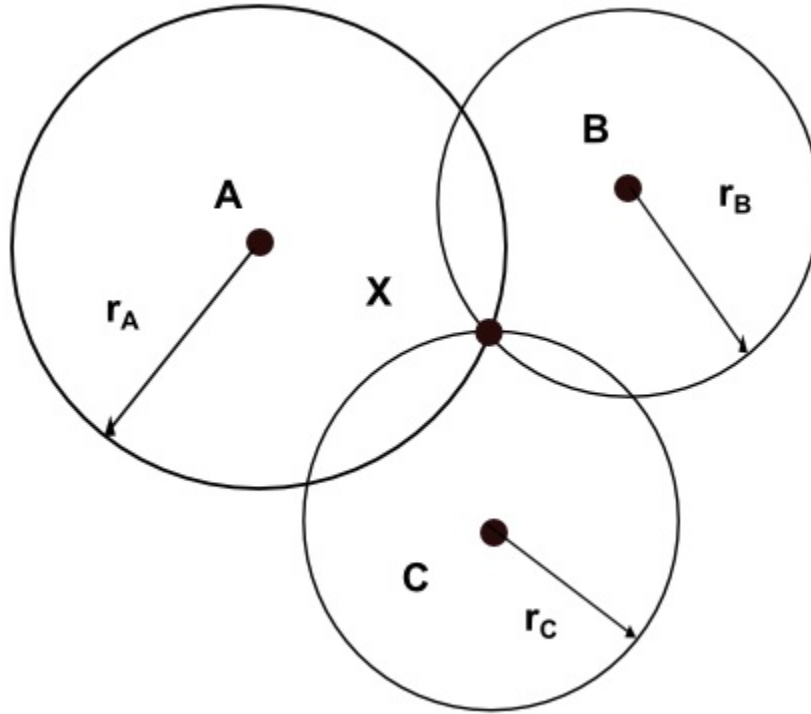


Figure 2.1: Tri-lateralization TOA.

One requirement of TOA is that the internal clocks in all the devices (including the transmitter) must be precisely synchronized. Given the high propagation speeds of transmitted signals, very small discrepancies in time synchronization can result in very large errors in location accuracy. TOA-based positioning solutions are typically challenging in environments where large amounts of multi-path, interference, or noise may exist. GPS is an example of a location system that uses TOA [1].

2.1.2 Time Distance of Arrival

TDOA uses relative time measurements instead of the absolute time measurements that TOA employs. While all the receivers need to be precisely synchronized in TDOA, it is not necessary for the transmitter to also be synchronized. Geolocation with TDOA

is performed through a process called hyperbolic-lateration [1]. Figure 2.2 illustrates this process.

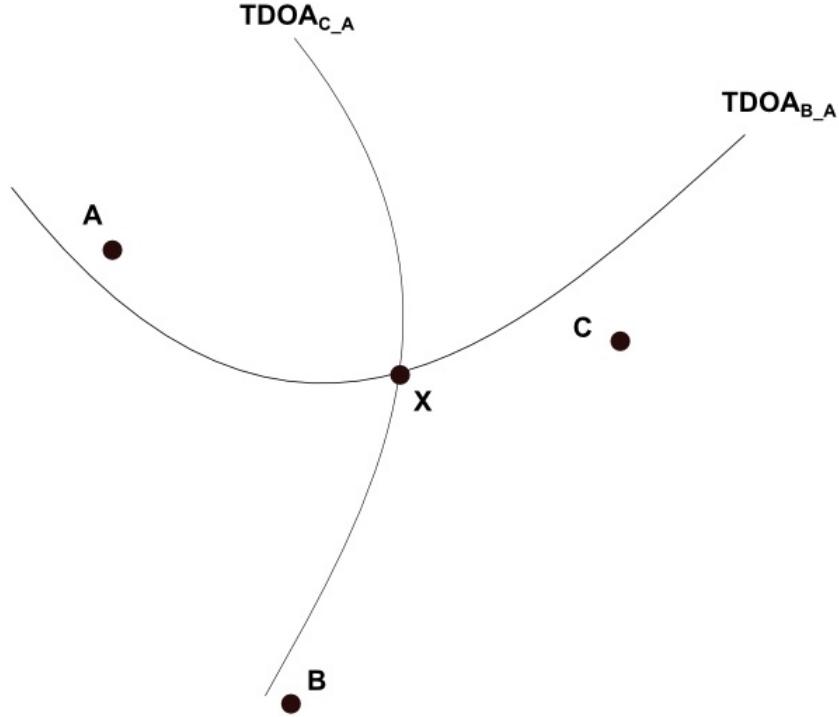


Figure 2.2: Hyperbolic-lateration TDOA.

Hyperbolas are drawn for each sensor pair. First, the constant TDOA is calculated for two sensors using Equation (2.1) [9]:

$$TDOA_{B-A} = |T_B - T_A| = k, \quad (2.1)$$

where $TDOA_{B-A}$ is the constant time difference between sensors A and B. It is calculated using T_A and T_B : the emitter's TOA to the two sensors respectively. The constant difference in time between the emitting signal's arrival to sensors A and B is k . The constant difference in distance is then calculated using Equation (2.2) [9]:

$$|D_{XB} - D_{XA}| = kc, \quad (2.2)$$

where D_{XB} and D_{XA} are the constant differences from the emitter to each sensor, respectively. The constant time difference k is multiplied to c , the speed of light (2.9×10^8 meters per second) in order to achieve units of meters. The hyperbola between the two sensors is drawn using Equation (2.3) [9]:

$$\sqrt{(x - x_A)^2 + (y - y_A)^2} - \sqrt{(x - x_B)^2 + (y - y_B)^2} = kc, \quad (2.3)$$

where x_A , y_A , x_B and y_B are the x-y coordinate locations of the two sensors respectively. The two variables in this equation are x and y which lie on the x-y coordinate plane and represent all possible x-y coordinates of the hyperbola. The hyperbola's foci are centered at the locations of sensors A and B. This process is repeated for each hyperbola. The point of intersection on the hyperbolas represents the estimated emitter position [9].

2.1.3 Angle of Arrival

AOA estimation uses a non-lateral approach to estimate the position of an emitter. It is accomplished through a phased-array antenna at each sensor that estimates the direction of the emitter's signal. The antenna is comprised of a sensor array and a real-time adaptive signal processor. The signal processor calculates and draws a Line-of-Bearing (LOB) for each sensor. The estimated position of the emitter is where the LOBs intersect. This process is called triangulation.

Only two receivers need to be employed to acquire a position estimate. Figures 2.3 and 2.4 convey how AOA is used to triangulate an emitter's position using two and three receivers, respectively.

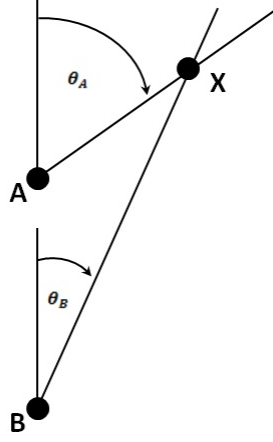


Figure 2.3: Triangulation using Two Receivers.

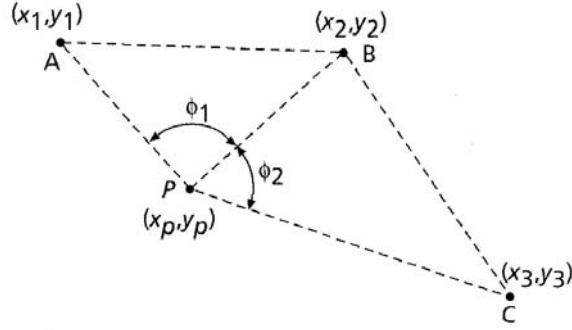


Figure 2.4: Triangulation using Three Receivers, used with permission [10].

The antennas in each sensor antenna array must be synchronized and require processing time to calculate the phase differences of the received emitter signal. However, no time synchronization is required between one AOA sensor and another. One application of AOA is the processing of radar signals [1].

2.1.4 Received Signal Strength

Geolocation can also be accomplished through RSS, which is defined as the voltage measured by a receiver's Received Signal Strength Indicator (RSSI). In this geolocation

method, sensors are arranged in a particular configuration in the network. Each sensor records the transmitted signal's RSS value in decibels (dBs), which is proportional to the logarithm of the distance from the emitter to the sensor. The RSS sensors report this value to a base station [1].

The base station collects the RSS values from all the sensors and inputs them into a position estimator such as a Maximum Likelihood Estimator (MLE) [1]. A very noisy position estimate can be determined with just three RSS sensors. As the number of RSS-reporting sensors in the network increases, the accuracy of the position estimate also increases.

RSS is a popular geolocation method because it is simple and inexpensive to implement. It should be noted however, that as the number of sensors used in the network increases, the cost to implement RSS geolocation increases. Additionally, position estimates can be unpredictable and highly inaccurate due to the variability in RSS measurements [1].

The task of geolocating an emitter on a military installation in real-time would require many sensors to cover the large area. Since RSS sensors are inexpensive, RSS is the most appropriate geolocation method for this research. In addition to being cheap, RSS sensors are easy to work with; they only measure one variable. TOA, TDOA and AOA sensors are significantly more expensive than RSS sensors.

The particular configuration that RSS sensors are arranged in will affect the accuracy of the estimator used to produce x-y position estimates of the emitter. Figure 2.5 displays three different sensor configurations used for simulated RSS geolocation scenarios. An MLE was used for each configuration to estimate the emitter positions. The results of the three configurations were compared to determine which configuration yielded the best accuracy.

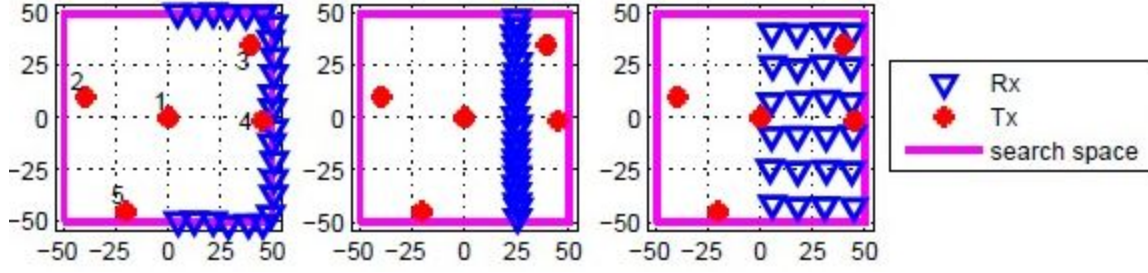


Figure 2.5: Example of Simulated RSS Geolocation Scenarios, used with permission [7].

The MLE was executed 1,000 times for each of the five emitters, generating 1,000 x-y position estimates for each emitter. Error ellipses for the Cramér-Rao Lower Bound (CRLB) and MLE were calculated and plotted for all five emitters for each sensor configuration. The CRLB presents the lower bound (or minimum) of an unbiased estimator's covariance (σ^2). In other words, the variance of an unbiased estimator must be greater than or equal to this bound. An estimator that achieves the CRLB is considered efficient. The MLE is a popular estimator and is unbiased. It is unbiased because on average, the estimated position of the emitter is correct. Additionally, it is known to consistently achieve the CRLB and has a Gaussian or normally-distributed Probability Density Function (PDF) [11].

The CRLB Confidence Interval (CI) ellipse bounds the area that 95% of the estimated emitter positions would reside in if the estimator used were as efficient as possible (if the minimum variance were achieved). The MLE CI ellipse bounds the area in which 95% of the estimated emitter positions actually resided in for the simulation. Figure 2.6 illustrates the x-y position estimate simulation results for five emitters and the 95% CI CRLB and MLE error ellipses for sensor configuration one. The MLE ellipse for emitter one was exactly the same as the CRLB for that emitter. The MLE ellipses for emitters two, three and four were slightly larger than the corresponding CRLB ellipses for those emitters. The MLE ellipse for emitter five was not completely accurate because the estimates were

confined by the search space. The search space effectively biased the estimator which resulted in an MLE ellipse that appeared tighter than it actually was.

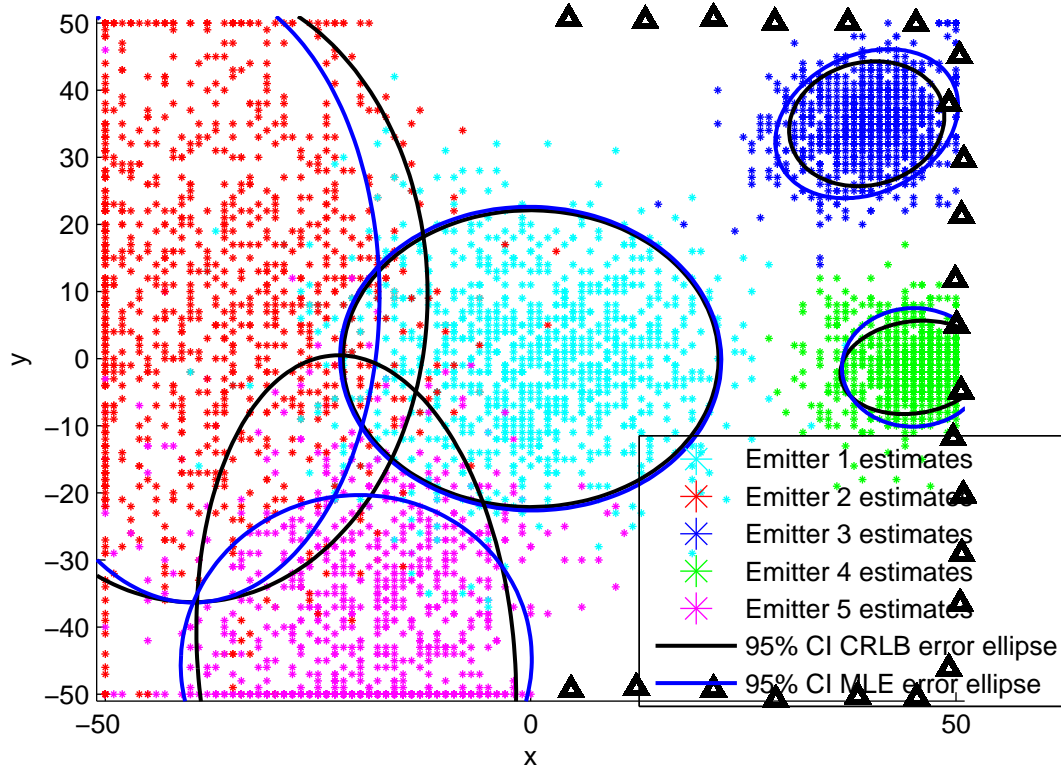


Figure 2.6: Emitter Estimates and 95% CI CRLB/MLC Error Ellipses for Sensor Configuration One [12].

Figure 2.7 illustrates the emitter estimates and error ellipses for sensor configuration two's simulation results. The MLE ellipses for emitters one, three and four extended significantly in both the positive and negative x direction when compared to the CRLB ellipse for those emitters. The MLE ellipse for emitter two was very similar to the CRLB ellipse. The MLE ellipse for emitter five was again biased by the search space. This configuration was the least ideal because the variances of estimated positions of all five emitters were high.

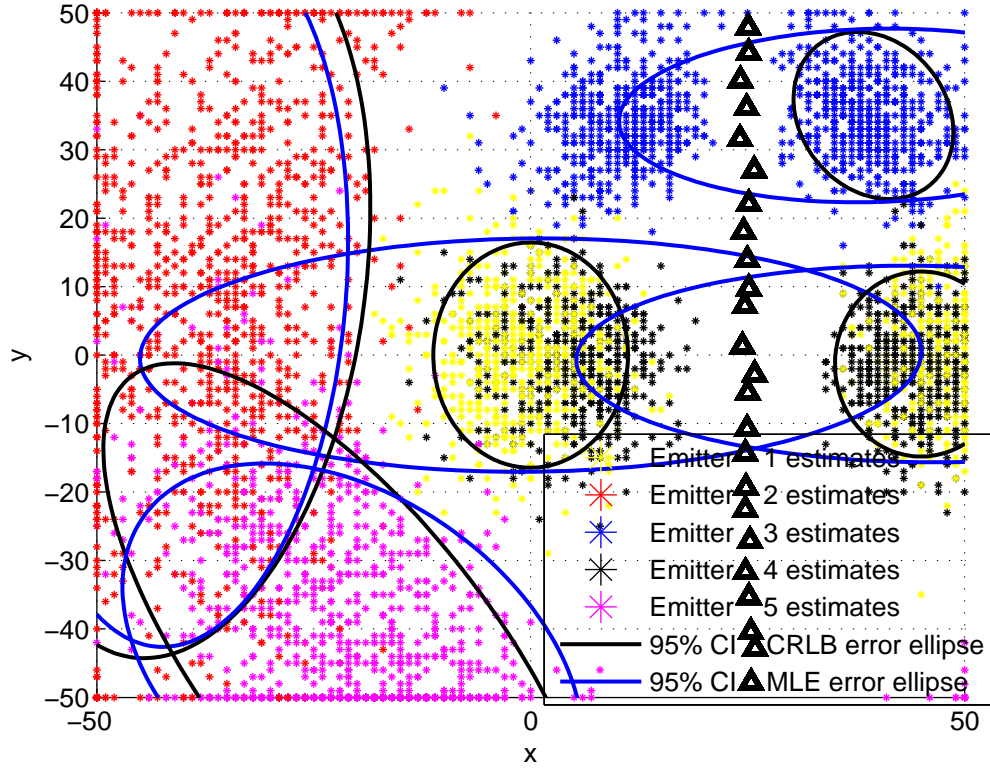


Figure 2.7: Emitter Estimates and 95% CI CRLB/MLC Error Ellipses for Sensor Configuration Two [12].

Figure 2.8 illustrates the emitter estimates and error ellipses for sensor configuration three. The MLE ellipses for emitters one, two, three and four were only slightly larger in area than their CRLB ellipse counterparts. The estimates for emitters one, three and four were very tight. The MLE ellipse for emitter four was also slightly biased by the search space. The MLE ellipse for emitter five was significantly biased by the search space.

Overall, sensor configuration three produced the tightest MLE ellipses for all five emitters. Emitters one, three and four had very tight ellipses. Emitter four's estimates varied significantly and subsequently had the largest CRLB and MLE ellipses. The farther

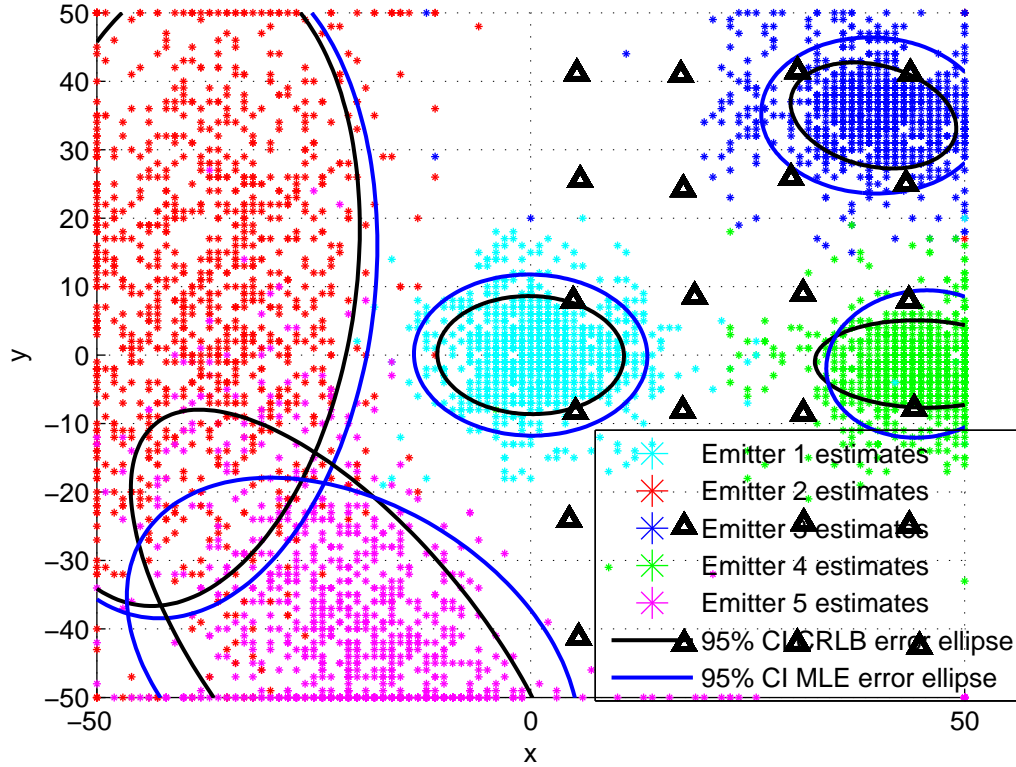


Figure 2.8: Emitter Estimates and 95% CI CRLB/MLC Error Ellipses for Sensor Configuration Three [12].

out of the sensor network the true emitter locations were, the higher the variance was of the estimates for those emitters.

This section presented a brief background on geolocation and four common localization techniques. An explanation was given to support why RSS is the most appropriate geolocation method for an RF emitter tracking and intent assessment. The results of simulated RSS geolocation scenarios for three different sensor configurations were presented and discussed. The next section will introduce the fundamentals of pattern recognition and its applications for determining the intent of an emitter in a WSN.

2.2 Pattern Recognition

Pattern recognition is an area of study in the Artificial Intelligence (AI) or machine learning research field and has many applications for speech recognition, image analysis and cognitive and computer science. Pattern recognition is the field of processing raw data and assigning that data to a certain class, or category. The four main stages of a pattern recognition system are: data collection, segmentation, feature generation, and classification [13]. All four stages are defined in this section and an explanation is given concerning how each stage applies to this research effort.

2.2.1 Data Collection

In the data collection (or sensing) stage, data from the subjects of interest are collected and stored [13]. For this research, Lat-Long position tracks were collected using the GPS unit. The position tracks were collected in real-time, and stored in a database used as training data for the classifier. The intent assessment in this research was designed to classify a position track as suspicious or non-suspicious in real-time.

2.2.2 Segmentation

In the segmentation stage, the collected data is differentiated from one subject to another. The segmentation process can be challenging if there are unclear or unestablished baselines to delineate one subject from another [13]. In this research, the intent assessment was designed to classify only one position track in real-time.

2.2.3 Feature Generation

In this stage, distinguishing features are generated from the subject's data set for classification. If more than one feature is generated from a subject data set, a feature vector is created for each subject [13]. In this research, five features were employed: dwell (or loiter time), repetition, deviation from roads and parking lots, proximity to a high-valued building, and proximity to a water tower. Subpixel detection was performed to determine

which pixel in the overhead image a position track resided in for each time stamp. Chapter III discusses the feature generation process for this research in detail.

2.2.4 Classification

The task of assigning an object to a class using the feature vector is defined as classification. A classifier that assigns a sample into one of two classes is formally called a dichotomizer but is more commonly known as a binary classifier [13]. A classifier that assigns an object into more than two classes is called a polychotomizer [13]. The classifier used in this research was a binary classifier because the class that each position track was assigned to was either suspicious or non-suspicious.

The degree of difficulty in classifying objects to the correct class is directly proportional to the variability of the feature generation data. The feature data for objects that belong to the same class can be varied due to complexity and noise. Bayesian decision theory is a common statistical process employed in pattern classification [13].

There are many pattern recognition methods for classifying subjects into classes. The four classification methods used to classify position tracks as suspicious or non-suspicious for this research were: the Linear Discriminant Function (LDF), the Diagonal Linear Discriminant Function (DLDF), the Quadratic Discriminant Function (QDF) and the Mahalanobis classification method. The next five subsections briefly describe certain classification methods.

2.2.4.1 LDA

LDA (or Fisher's Linear Discriminant) is a classification method that uses supervised training to reduce an ' n ' dimensional data set of a two-class classification problem to one dimension [13]. Multiple Discriminant Analysis (MDA) is an extension of LDA applied to a multiple class problem. MDA reduces an ' n ' dimensional data set to a $C - 1$ dimensional data set where C is the number of classes [13].

2.2.4.2 LDF

It was determined that MATLAB®'s classify function uses discriminant functions to classify data. LDFs classify a data set by determining the optimal separating hyperplane. Equation (2.4) is the general function of an LDF:

$$f(x) = \mathbf{w}^T \mathbf{x} + w_0, \quad (2.4)$$

where \mathbf{w} is the weight vector, \mathbf{x} is the input vector (the data sample of an object to be classified into a certain class) and w_0 is the bias or threshold. The input vector \mathbf{x} is classified according to the conditional statements below:

$$\mathbf{x} \in C_1, \text{ if } f(\mathbf{x}) \geq 0,$$

$$\mathbf{x} \in C_2, \text{ if } f(\mathbf{x}) < 0,$$

where C_1 and C_2 denote class one and two respectively. The hyperplane is a $d - 1$ dimensional surface, where d is the number of features of the data [13]. In this research, d was five and was explained in subsection 2.2.3. The weight vector \mathbf{w} is easily determined by use of the matrix form of Equation (2.5) as [13]:

$$\mathbf{X}\mathbf{w} = \mathbf{b}, \quad (2.5)$$

where \mathbf{X} is the matrix of training data and \mathbf{b} is an arbitrary user-defined vector. Equation (2.6) shows (2.5) in expanded form.

$$\begin{bmatrix} x_{10} & x_{11} & \cdots & x_{1d} \\ x_{20} & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n0} & x_{n1} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (2.6)$$

In (2.6) [13], d is the number of features of the data and n is the number of samples included in the training data set which was 22 for this research. There were 22 position tracks included in the position track database. There are $d + 1$ columns in the training data matrix (X) of (2.6).

In X , the samples are rows and the features are columns. The dimensions of X were 22 x 6 for this research. The dimension of w was 6 x 1 to include the bias (w_0). The dimension of b was 22 x 1. There are many possible solutions for b . One possible solution is that all rows in the matrix are ones.

From (2.5), if X is non-singular (or invertible), X^{-1} can be multiplied to both sides of the equation, as shown in Equation (2.7). Non-singular matrices have an inverse that exists and can be calculated as [13] :

$$XX^{-1}w = X^{-1}b. \quad (2.7)$$

Equation (2.8) can then be used to solve for w as [13] :

$$w = X^{-1}b. \quad (2.8)$$

However, X is typically singular (which means that the inverse of X does not exist) [13]. While X^{-1} cannot be solved directly, the hyperplane can still be determined. First, X^T is multiplied to both sides of the equation as shown in Equation (2.9) [13].

$$X^T X w = X^T b \quad (2.9)$$

Then, $(X^T X)^{-1}$ is multiplied to both sides of the equation as shown in Equation (2.10) [13].

$$(X^T X)(X^T X)^{-1}w = (X^T X)^{-1}X^T b \quad (2.10)$$

Equation (2.10) quickly simplifies because $(\mathbf{X}^T \mathbf{X})^{-1}$ cancels the $(\mathbf{X}^T \mathbf{X})$ on the left side of the equation. Equation (2.11) is the resulting equation solved for \mathbf{w} .

$$\mathbf{w} = \underbrace{(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T}_{\text{pseudoinverse}} \mathbf{b} \quad (2.11)$$

The product $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ is called the pseudoinverse and is multiplied to \mathbf{b} to determine \mathbf{w} . The bias (w_0) of the LDF function can be calculated when the value of $f(x)$ is 0. This is demonstrated in Equation (2.12) [13].

$$0 = \mathbf{w}^T \mathbf{x} + w_0 \quad (2.12)$$

Equation (2.13) is the result when w_0 is subtracted from both sides and then division is performed by $\|\mathbf{w}\|$ [13], which is the l_2 norm of the hyperplane.

$$\frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|} \quad (2.13)$$

When \mathbf{w} and w_0 are determined, any input vector \mathbf{x} can be classified into the appropriate class using the discriminant function in (2.4) [13].

2.2.4.3 DLDF

The DLDF classification method is similar to the LDF method except that a diagonal covariance matrix is developed. This matrix asserts that the five features used in this research were independent. A classifier that has a diagonal covariance matrix is called a naive bayes classifier [13].

2.2.4.4 QDF

QDF is simply an extension of the LDF but instead uses a quadratic function. Equation (2.14) is the general form for the quadratic discriminant function.

$$f(x) = \mathbf{w}^T \mathbf{x}^2 + \mathbf{w}^T \mathbf{x} + w_0 \quad (2.14)$$

In the same manner as (2.4), \mathbf{w} is the weight vector, \mathbf{x} is the input vector and w_0 is the bias.

2.2.4.5 Mahalanobis Method

The Mahalanobis method was the fourth classification method employed in this research. It is fundamentally different than the previous three classification methods. This method uses a distance metric to classify samples into classes. It makes use of the fact that the direction of a data set's variance plays an important role in classification [13]. For example, the standard deviation of the distribution of data could resemble an ellipse, as shown in Figure 2.9 [13].

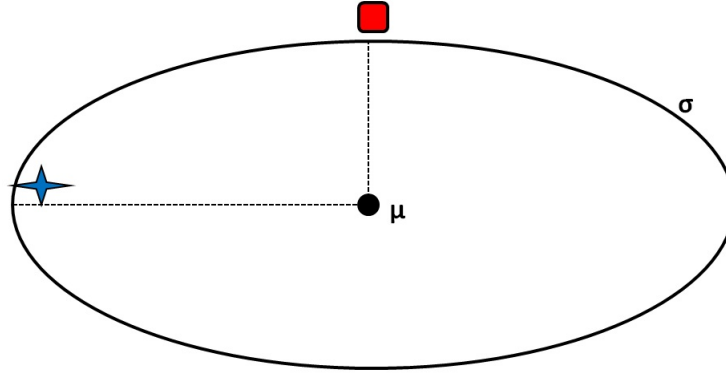


Figure 2.9: Ellipse-shaped distribution of data represented by the standard deviation (σ) .

This method can identify outliers in the data set by accounting for the standard deviation (σ) of the data [13]. In Figure 2.9, the blue star data-point has a greater euclidean distance from the data's mean (μ) than the red square; however, the red square data point is outside the standard deviation (σ) of the data and thus can be considered an outlier. Equation (2.15) is the general equation for the Mahalanobis method [13]:

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})}, \quad (2.15)$$

where \mathbf{x} is the multivariate input vector, $\boldsymbol{\mu}$ is a vector of means, \mathbf{S} is the covariance matrix and $D_M(\mathbf{x})$ is the Mahalanobis distance vector for \mathbf{x} . The MATLAB® documentation for the classify function states that Mahalanobis distances are calculated with stratified covariance estimates.

2.2.4.6 Classifier Terminology

The classifier will always classify an object into one of the classes defined by the problem. In this research, a position track is always classified as either suspicious or non-suspicious. If the classifier fails to classify the object into the correct class, a false positive error (or Type I error) or a false negative error (Type II error) occurs [14]. These errors are now explained in the context of this research. First, the suspicious class of position tracks is now referred to as H_1 and the non-suspicious class of position tracks is referred to as H_0 . A false positive error occurs when the classifier classifies a non-suspicious position track as suspicious. In mathematical terms, H_0 is classified as H_1 . A false positive error is also called a ‘false alarm.’

Conversely, a false negative error occurs when the classifier incorrectly classifies a suspicious position track as non-suspicious. In mathematical terms, H_1 is classified as H_0 . A false negative error is also called a ‘miss.’ The consequences of a false negative error are more severe than a false positive error because security personnel would not engage or investigate a suspicious track mis-classified as non-suspicious. A compromise of the military installation could result.

The probability or rate that a false positive error or false alarm occurs is expressed mathematically as P_F and is defined as $P[H_1|H_0]$. This expression reads, “the probability of H_1 given H_0 .” It is the probability that the classifier selects a position track as suspicious given that it was non-suspicious. The probability or rate that a false negative error or miss occurs is expressed mathematically as P_M and is defined as $P[H_0|H_1]$. This expression

reads, “the probability of H_0 given H_1 .” It is the probability that the classifier selects a position track as non-suspicious given that it was suspicious.

When a classifier correctly classifies a suspicious track as suspicious it is referred to as a ‘detect.’ The probability of detection is expressed as P_D and is defined $P[H_1|H_1]$. P_D is inversely proportional to P_M and is expressed in Equation (2.16).

$$P_D = 1 - P_M \quad (2.16)$$

When a classifier correctly classifies a non-suspicious track as non-suspicious it is referred to as a ‘reject.’ The probability of rejection is expressed as P_R and is defined $P[H_0|H_0]$. P_R is inversely proportional to P_F and is expressed in Equation (2.17).

$$P_R = 1 - P_F \quad (2.17)$$

2.2.4.7 Reporting Methods

There are multiple methods to report the results of the classifier. The accuracy, P_M and P_F can be graphed as a function of a sweeping parameter that is incremented from one numeric value to another. The two parameters that were swept to generate classifier statistics in this research were the grid cell width parameter (w_{cell}) and the prior probability of the suspicious class ($P[H_1]$).

Receiver Operating Characteristic (ROC) curves can be generated which display P_D vs. P_F data. ROC curves communicate how well the classifier correctly classifies an object that is a member of the class H_1 into the H_1 class, compared to how often it classifies an object that is a member of the class H_0 into the H_1 class. Chapter IV presents the classifier statistics for this research through the two parameter sweeps discussed in the previous paragraph and P_D vs. P_F data which was used to generate best fit ROC curves.

This section presented the fundamentals of pattern recognition. The four main stages of this field were defined and applied in the context of this research. In the classification

subsection, the four classification methods employed in this research were introduced, and equations were presented for them. Classifier terminology that is used in this thesis was then discussed. Finally, the different ways of reporting the classifier statistics were communicated. The next section will discuss the behavior classification process of an RF emitter in a WSN.

2.3 Pattern Recognition Applied to WSNs

As stated at the beginning of this chapter, one solution to increasing the physical security for a delineated area such as a military installation is to employ pattern recognition techniques in WSNs to detect anomalous events. In section 1.1, it was explained that pattern recognition techniques can be applied to geolocated position tracks produced by WSNs to determine if an emitter in motion is malicious or not. This process is referred to as anomaly detection or intent assessment. Anomaly detection is a subset of the behavior classification field in which patterns are classified into two classes: suspicious (or malicious) and non-suspicious [15]. Figure 2.10 illustrates the flow from collection of surveillance target data in a sensor network to intent classification.

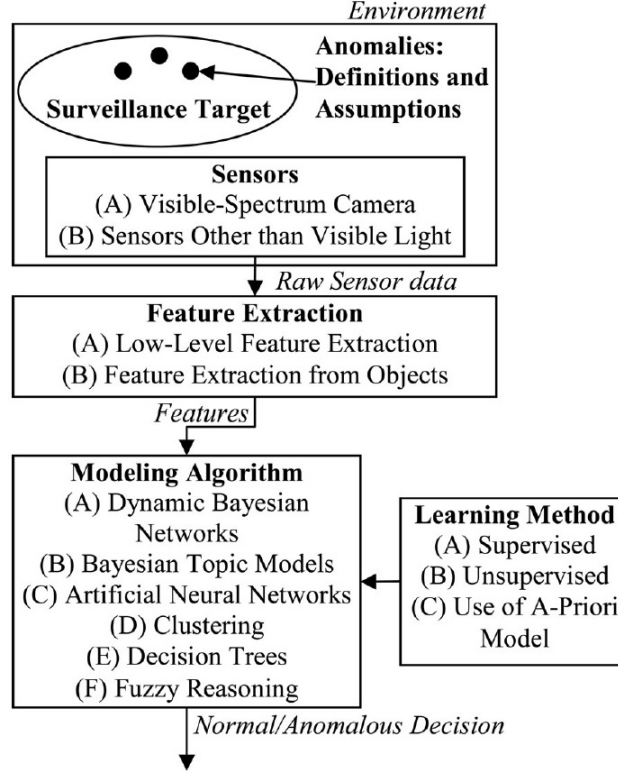


Figure 2.10: Diagram of the Flow From a Physical Environment to Anomaly Detection, used with permission [15].

By employing a geolocation method discussed in section 2.1 in real-time, an estimated position track can be generated and recorded to classify the behavior. RF devices that measure TOA, TDOA, AOA or RSS are examples of sensors that could be used in this model. The five features generated were introduced in subsection 2.2.3: dwell (or loiter time), repetition, deviation from roads and parking lots, proximity to a high-valued building, and proximity to a water tower.

The modeling algorithms used in this research were the four classification methods discussed in subsection 2.2.4: LDF, DLDF, QDF and the Mahalanobis method. The learning method used in this research was supervised, because each position track in the database was determined beforehand, during the data collection stage as non-suspicious

or suspicious. Finally, the decision of normal or anomalous behavior at the end of the flow chart is equivalent to the output of the intent assessment system in this research. The decision terminology for the two classes in this research's assessment system were suspicious and non-suspicious.

This section presented a top-level view of the flow from passive surveillance of an RF emitter in a WSN to an output decision that assessed the emitter's intent. The next section will discuss current research techniques in the area of applying pattern recognition in WSNs to detect malicious activity in the cyber domain.

2.4 Pattern Recognition Applied to Cyber Security

Research into anomaly detection within a WSN is also being conducted in the computer network domain as well as the physical domain. Pattern recognition techniques can be employed in sensor networks to detect anomalous computer network activity. This activity could result in a cyber or computer network-based attack. Neural Networks (NNs) are one pattern recognition tool that can be used to detect cyber attacks [16]. NNs are composed of interconnected neurons or nodes that are used to solve AI problems.

One example of a cyber attack is a Distributed Denial of Service (DDoS) attack [4]. DDoS attacks consist of a large number of network service requests towards a victim node. A DDoS computer network attack can be realized in a WSN by attacking a target sensor with the intent of exhausting the energy resources available to them. The targeted sensor is then incapable of performing further sensing operations. Detection of DDoS attacks in a WSN can be accomplished using a Graph Neuron (GN) NN [4]. The algorithm performs comparisons of current network traffic patterns to a database of normal sensor network traffic [4].

Electrical power grids and substations are examples of infrastructures that can be highly vulnerable to cyber attacks [16]. Research has thus been conducted to specifically prevent these kind of attacks. A cyber attack on a power substation could have the intent

of disrupting or denying the power supply to commercial or residential areas. Figure 2.11 illustrates a Probabilistic Neural Network (PNN) designed for detecting electrical faults caused by cyber attacks in a power substation. A PNN is a type of NN called a feedforward NN. In a feedforward NN, information always flows forward through the network, as opposed to forwards and backwards [16].

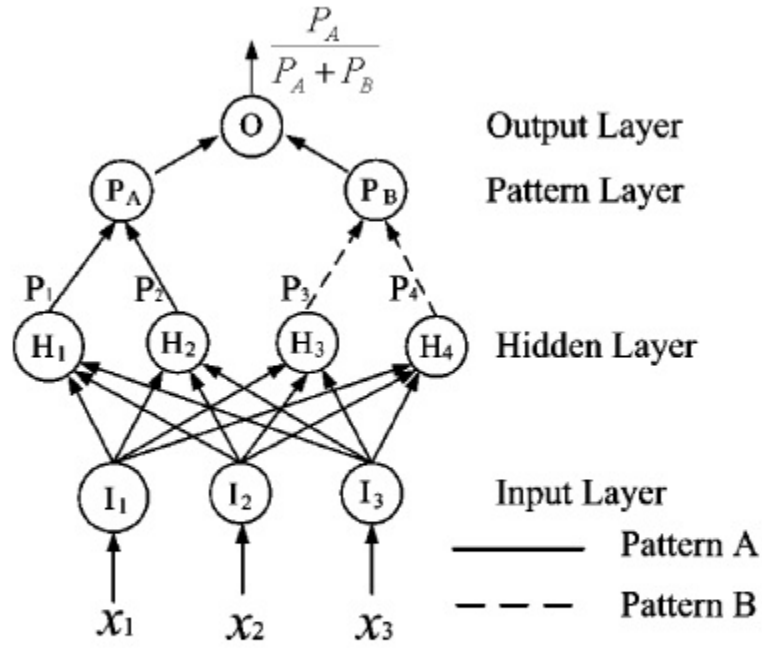


Figure 2.11: Architecture of a PNN Used for Detecting Authentic Faults in Power Substations, used with permission [16].

The input vector \mathbf{X} ($[x_1 \ x_2 \ x_3]$) is called the testing exemplar. The testing exemplar elements x_1 , x_2 and x_3 are voltage readings taken from different locations in the substation. P_A and P_B represent the probabilities of a real fault and fake fault, respectively. If $P_A/(P_A + P_B) > 0.5$, the exemplar data is classified into class A, or a real fault. If $P_A/(P_A + P_B) < 0.5$, the exemplar data is classified into class B (a fake fault). A fake fault is a possible indication that a power substation's fault protection system has been compromised by a cyber attack [16].

In this section, DDoS attacks were defined and pattern recognition techniques that use WSNs to detect malicious cyber activity were introduced. The top-level architecture of a PNN designed to classify power substation data into real and fake faults was briefly covered.

2.5 Chapter Summary

This chapter outlined principles of both source localization and pattern recognition. Source localization was defined and the principles of TOA, TDOA and AOA were covered briefly. Geolocation using RSS was discussed in greater and detail and the results of RSS simulations of five emitters in three different sensor configurations using an MLE were presented.

Pattern recognition was then defined and the four main stages of a pattern recognition system were described in detail. Each stage was explained in the context of this research. In the classification stage, equations were provided for the four classification methods used in this research, with an emphasis on the LDF derivation. Additionally, classifier terminology was introduced and methods of reporting classifier statistics were covered.

A brief overview was presented in the area of employing pattern recognition techniques in a WSN to detect anomalous behavior in the physical domain. Finally, current research that employs pattern recognition in WSNs to identify computer network (or cyber)

attacks was discussed. Chapter III applies the theory that was presented in chapter II towards the methodologies used in this research.

III. Research Methodology

This chapter describes the methodologies performed to collect and process x-y position tracks using the Magellan[®] Mobilemapper GPS unit, generate features from the collected position tracks and process the generated feature data to perform intent classification on a single position track in MATLAB[®]. A database was created of position tracks collected from the GPS unit which was used by the classifier. This database was comprised of position tracks intentionally created to replicate suspicious and non-suspicious activity on a military installation. Figure 3.1 presents the specific GPS unit used to collect the position tracks in this research.

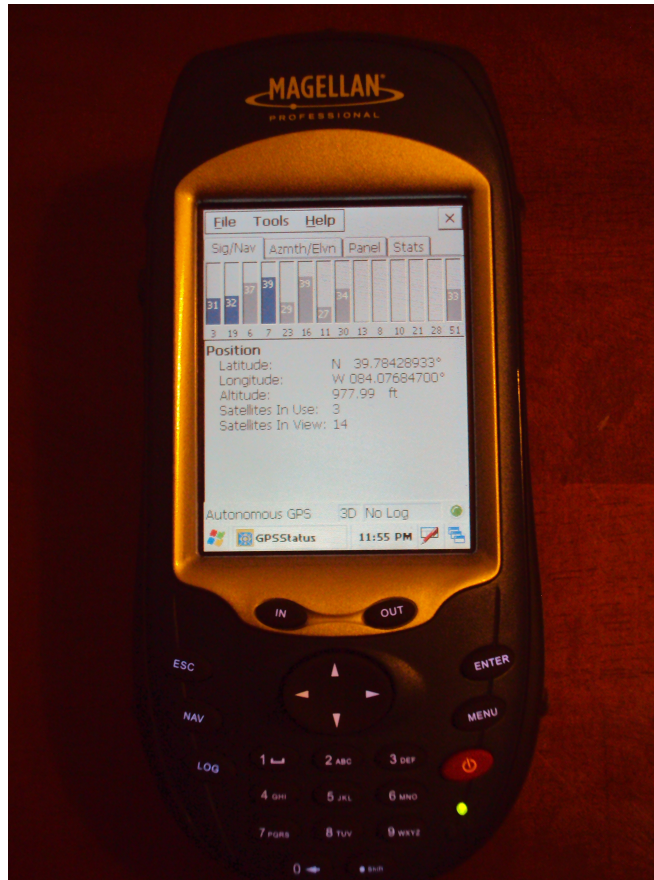


Figure 3.1: Magellan[®] Mobile GPS Unit in Autonomous GPS Mode.

3.1 Magellan[®] GPS Unit Operation

The Magellan[®] GPS device user interface was straightforward and simple to operate. When the unit was acquiring and maintaining communication with the GPS satellites, it occasionally alternated between the autonomous and differential GPS modes. This fluctuation was due to the variability of the GPS constellation's signal strength in different conditions. The GPS unit's log functionality enabled real-time acquisition of Lat-Long coordinates which were stored over time to create a position track. Each position track recorded by the GPS unit was collected within the physical area displayed by Figure 3.2.

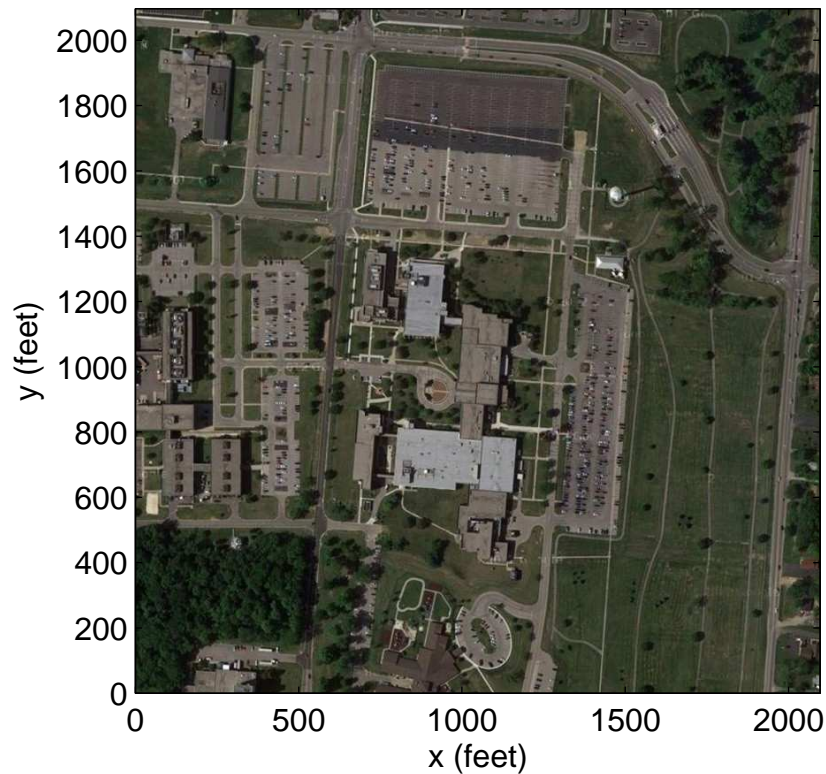


Figure 3.2: Overhead Image of Position Track Collection Area.

This image is a section of Area B, WPAFB, and was captured using Google Maps[®]. The dimensions of this image were 776 x 776 pixels. Google Maps[®] was also used to convert the x and y axes from pixel units to feet. To accomplish this, two points at about the same pixel row were picked on the overhead image plotted in MATLAB[®]. MATLAB[®] was then used to zoom in on the two points and the distance between the two points in pixels was counted and recorded. The distance scale in Google Maps[®] was then used to measure the distance in feet from the same two points.

It was determined through dimensional analysis that one pixel had a distance of 2.7 feet. Since the dimensions of the overhead image were 776 x 776 pixels, the dimensions of the image in feet were $(776 * 2.7) \times (776 * 2.7)$ or 2,095 feet x 2,095 feet. The imagesc

command in MATLAB[®] was used to correctly scale the x and y axes. The next section describes how the position tracks were imported into MATLAB[®] so feature generation could be performed on them.

3.2 Position Track Processing

This section describes the process of importing the position tracks from the GPS unit into MATLAB[®] and converting the Lat-Long coordinates into x-y coordinates for feature generation.

3.2.1 Importing GPS Data into MATLAB[®]

Each position track text file was ported into a personal computer workstation through a Universal Serial Bus (USB) connection. A function extracted the Lat-Long data from the National Marine Electronics Association (NMEA) data format that was stored by the GPS unit. The Lat-Long data was stored in a $L \times 2$ matrix where L was the length or total number of time-stamps of a particular track. The two columns encompassed the latitude and longitudinal numeric values, respectively. A third column was added to the matrix that stored the time-stamp for each Lat-Long entry. This column started with a numeric value of one and was incremented by one to L (the last set of Lat-Long coordinates of a position track had a time-stamp of L).

3.2.2 Lat-Long to x-y Coordinate Conversion

It was then necessary to convert the $L \times 3$ time-stamp-Lat-Long matrix for each position track to a matrix of x-y coordinates corresponding to the locations on the overhead imagery map in Figure 3.2. A function converted each Lat-Long time-stamp to an x-y coordinate pair. It was necessary to determine the Lat-Long coordinate pair of the bottom-left corner of the overhead image acquired from Google Maps[®]. The origin of the overhead image was at the bottom-left corner and had the x-y coordinates (0,0). The newly created time-stamp-coordinate matrices also had the dimensions $L \times 3$ for each position track. The

first column of the matrix encompassed the individual time-stamp. The second and third columns stored the x and y coordinate values respectively.

Figure 3.3 illustrates a processed position track that was collected using the GPS unit. The track is green because it was collected with the intent of modeling non-suspicious behavior.

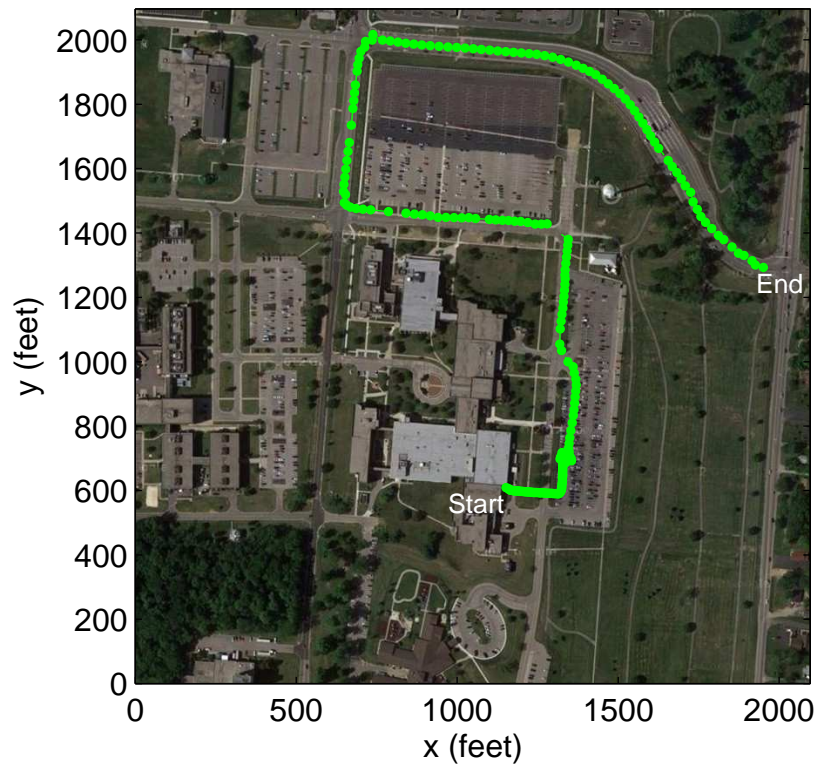


Figure 3.3: An Example Position Track Collected with the Magellan® GPS Unit.

The 22 position tracks in the position track database were all processed before feature generation on them was performed. The next section explains the five feature generation algorithms that were performed on the position tracks.

3.3 Feature Generation

This section describes the processes employed to generate feature data from the position track database. The five features used in this research were first introduced in subsection 2.2.3 and are listed again: dwell (or loiter time), repetition, deviation from roads and parking lots, proximity to a high-valued building, and proximity to a water-tower. The dwell and repetition feature algorithms required the use of a variable resolution grid which is discussed in subsection 3.3.1. The remaining three features belonged to a category of features called landmark features. The algorithms for these features required a unique landmark distance map. The development of the landmark distance maps is explained in subsection 3.3.4.

3.3.1 Development of Grid

It was necessary to create a 2-dimensional grid in MATLAB[®] to generate data for the dwell and repetition features. The grid itself was square in shape, and partitioned the 776 x 776 pixel overhead image into square-shaped cells of equal area. The resolution of the grid could be adjusted, which altered the total number of cells within the grid. The grid was used to keep track of which cell a position track resided in for a given time stamp. A significant amount of book-keeping was required for this. Once the dwell and repetition feature algorithms were able to determine which cell a position track was in, code could be written to determine when a position track left the current cell and when (if ever) it returned to the current cell and how many times. Figure 3.4 illustrates a grid configuration with a grid cell width w_{cell} of 81 feet.

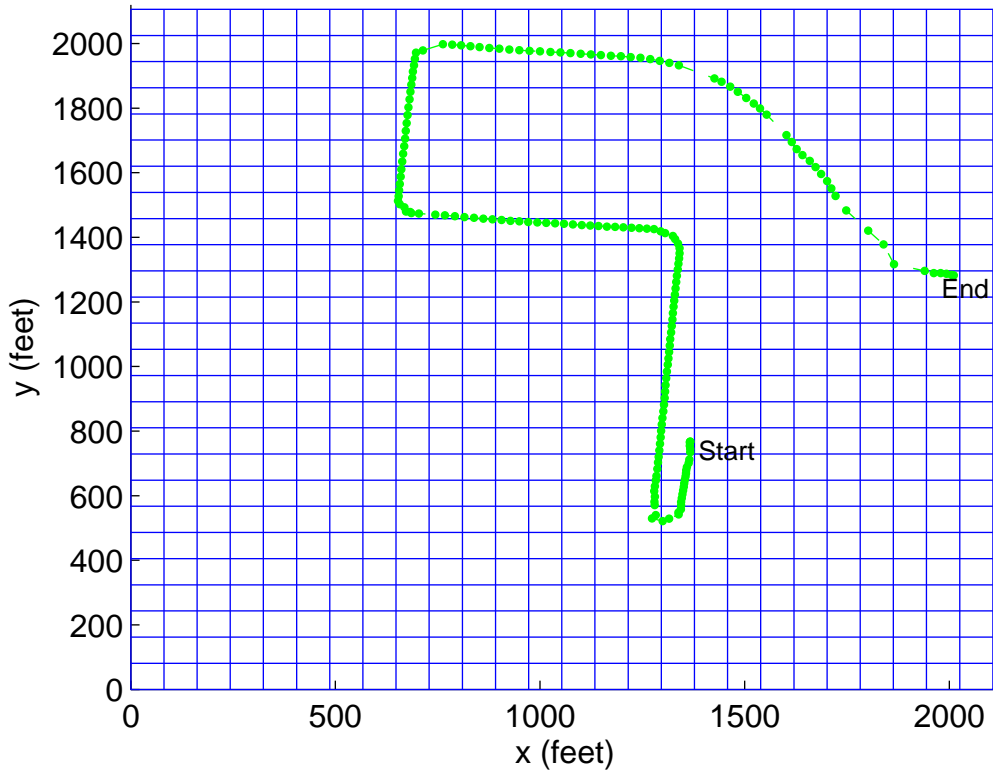


Figure 3.4: Variable Resolution Grid Developed for the Dwell and Repetition Features: $w_{cell} = 81$ feet.

Subsections 3.3.2 and 3.3.3 will discuss how the grid designed for this research was used to generate feature data for the dwell and repetition features.

3.3.2 *Dwell*

The first feature used to classify position tracks as suspicious or non-suspicious was the dwell time (or loiter) feature. A numeric score for this feature was determined by counting how many time stamps a certain position track remained in the same grid cell. This was accomplished by traversing through position track starting at time-stamp one. The x and y coordinates for each time-stamp were accessed and used to determine which exact cell the position track was currently residing in. A local MATLAB® variable incremented every time a position track remained in the same cell, as the next time-stamp was accessed.

A segment of MATLAB® code is included to illustrate how the dwell algorithm was implemented.

```
max_count = 0;
x_est = track(i,2); y_est = tracks(i,3);
% determine which cell the estimate is in
cell_x = ceil(x_est/grid_spacing);
cell_y = ceil(y_est/grid_spacing);
prev_x = cell_x; prev_y = cell_y;
for i = 2:length(track)
    x_est = track(i,2); y_est = tracks(i,3);
    % determine which cell the estimate is in
    cell_x = ceil(x_est/grid_spacing);
    cell_y = ceil(y_est/grid_spacing);
    if cell_x == prev_x & cell_y == prev_y
        % track stayed in the same cell
        count = count + 1;
        if count > max_count
            max_count = count;
        end
    else % track moved to new cell
        count = 0;
        prev_x = cell_x; prev_y = cell_y;
    end
end
end
```

Figure 3.5 illustrates a collected position track that generated a low dwell feature score.

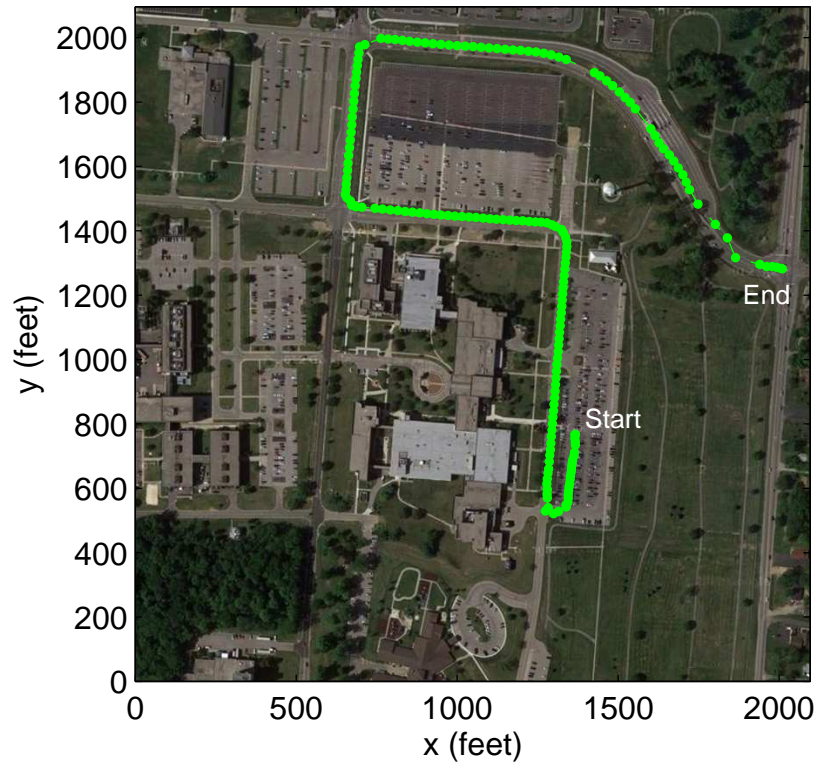


Figure 3.5: Position Track with Low Dwell Time.

Figure 3.6 illustrates a collected position track that generated a high dwell feature score. The track is red because it was collected with the intent of modeling suspicious activity.

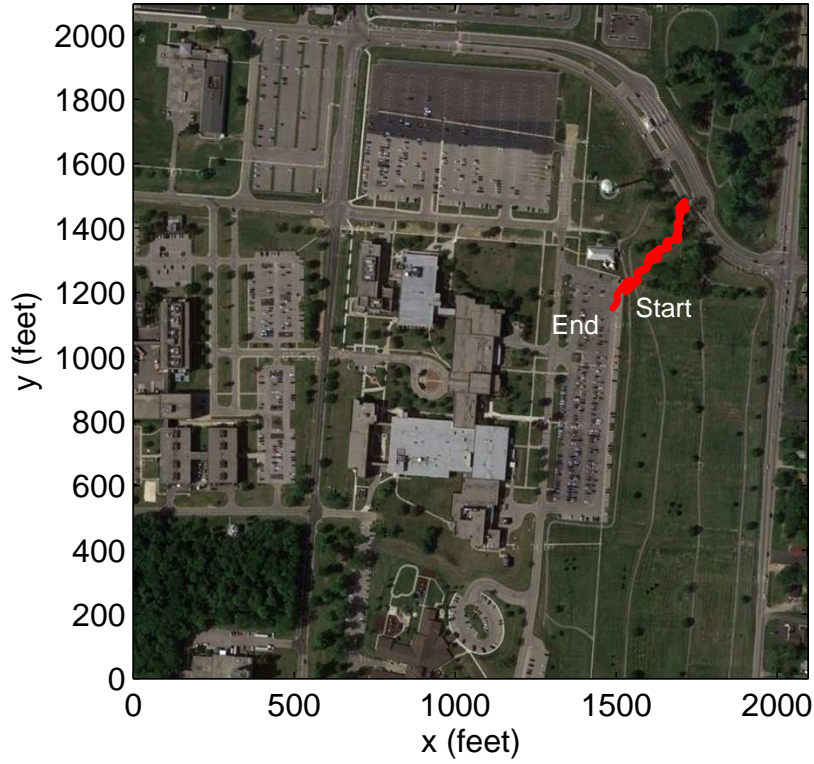


Figure 3.6: Position Track with High Dwell Time.

This subsection described how the dwell time feature data was generated for the 22 position tracks. Procedural code was inserted to illustrate specifically how the algorithm operated. Additionally, two position tracks were shown that had low and high dwell time scores. The next subsection describes the development of the repetition feature algorithm.

3.3.3 Repetition

The repetition feature was employed to determine if a position track exhibited repetitive activity within the data collect area. In the context of this research, repetition was defined as the number of times a position track returned to the same grid cell. The expression used to generate a score for repetitive position track behavior was:

$$\sum_{c=1}^N n_c^2, \quad (3.1)$$

where N was the total number of grid cells in the current resolution configuration, c was the current grid cell and n_c was the number of times that a particular position track returned to the current grid cell c during its collect. The exponent was a penalty factor and was set to two for this research. Figure 3.7 illustrates a collected position track that exhibited low repetitive activity.

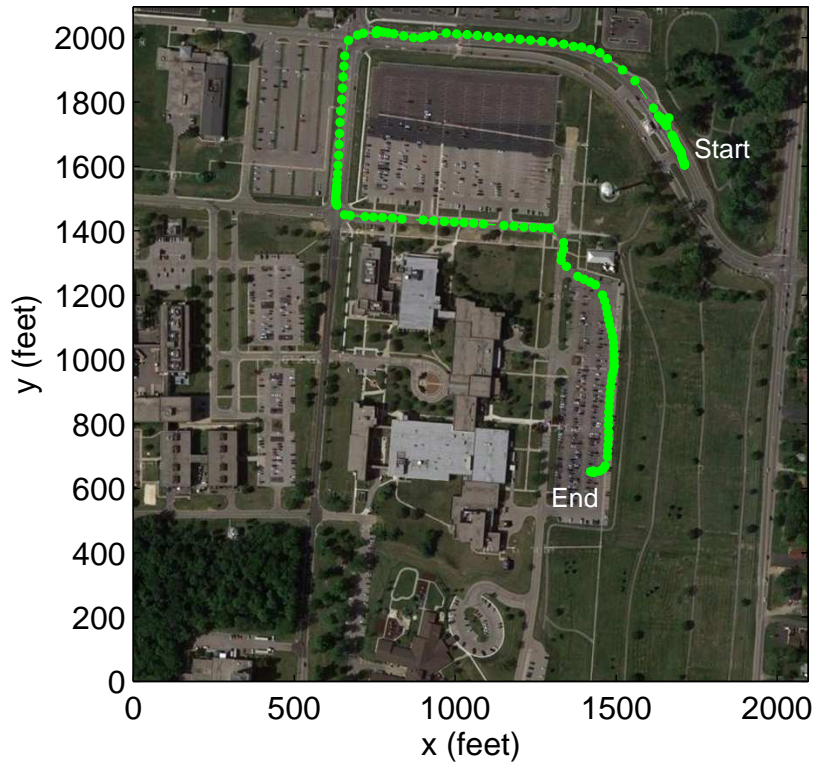


Figure 3.7: Position Track with Low Repetition.

Figure 3.8 illustrates a collected position track that exhibited high repetition.

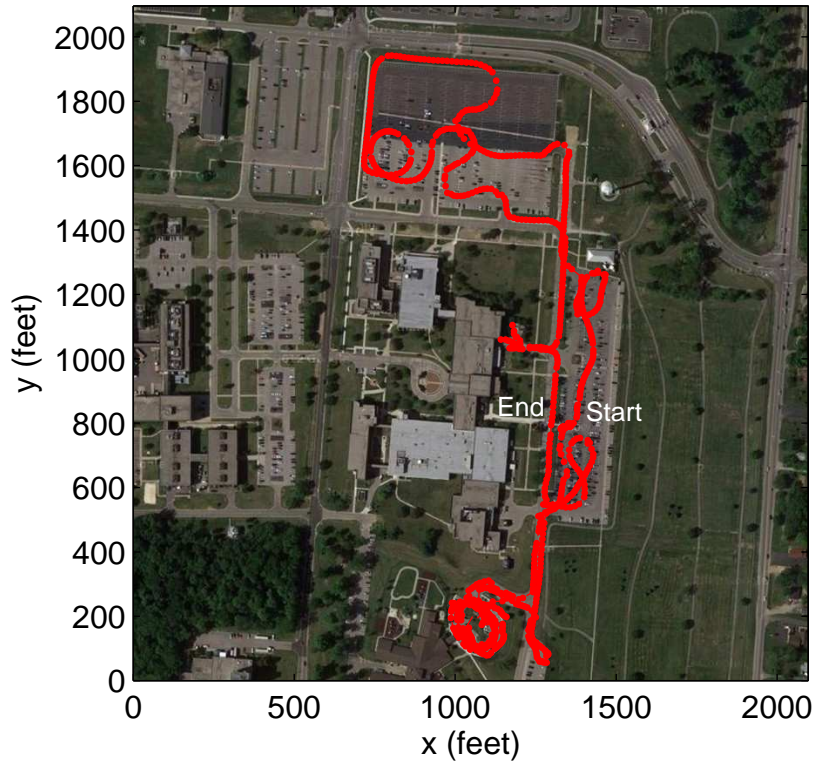


Figure 3.8: Position Track with High Repetition.

This subsection described how repetition feature data was generated for the 22 position tracks. Equation (3.1) expressed the formula to calculate a repetition score for each position track. Additionally, two position tracks were presented that had low and high repetition scores. The next subsection describes the development of the landmark distance maps for the three landmark features.

3.3.4 Development of Landmark Distance Maps

As stated in the beginning of this section, it was necessary to develop distance maps in MATLAB® for the three landmark features used in this research effort. This was accomplished by first creating a bitmap image that highlighted the areas of interest for each landmark. The dimensions of the bitmap image were the same as the data collection area

overhead image presented in Figure 3.2: 776 x 776 pixels. For the deviation from roads and parking lots feature, pixels that encompassed either a road or parking lot were highlighted. For the proximity to high-valued building and water tower features, the bitmaps highlighted the pixels that encompassed these landmarks.

For the high-valued building and water tower bitmaps, it was relatively simple to manually highlight the individual pixels that comprised those landmarks. However, the task of highlighting all the areas in the overhead image that occupied road or parking lot occupied proved to be a formidable task. The most efficient way this was accomplished was by using MATLAB® to search the original overhead image for pixels that had the same shading as pixels that were comprised of a road or parking lot. The deviation from roads and parking lots bitmap was created this way. The three bitmaps created all had dimensions of 776 x 776 pixels.

After the three bitmaps were created, the distance maps for the three landmark features were created by a function in MATLAB®. This function inputted a 776 x 776 pixel bitmap and returned a 776 x 776 pixel image for the given landmark feature. The function searched the bitmap for the pixels that had values of zero. These pixels corresponded to the highlighted areas of a given landmark feature. Then, pixels adjacent to the zero-valued pixels were assigned a value of one because they were one pixel (or one unit of distance) away from the landmark feature area.

Then, non-zero-valued pixels that were adjacent to the pixels that had values of one were set to two because they were two pixels (or two units of distance) away from the landmark feature area. This process continued until every pixel in the 776 x 776 map that was not part of the highlighted area (and originally set to zero) was assigned a non-zero value. In this way, a sophisticated landmark distance map was created for each of the three landmark features used in this research.

In the case of the roads and parking lots distance map, if a pixel was 200 units away from a certain parking lot but only 100 units away from a road, the numeric value of 100 was stored to the pixel. The function that created the landmark distance maps ensured that any pixel in the map that was not a member of the highlighted area was set to the minimum distance away (or closest distance to) the nearest pixel belonging to the highlighted area.

This subsection explained how the landmark distance maps were created in MATLAB®. The next three subsections describe the processes used to generate feature data using the three landmark features. Figures of each landmark distance map that were used in this research are presented.

3.3.5 Deviation from Roads and Parking Lots

This feature reported the maximum value that a position track deviated from either a road or a parking lot within the confines of the overhead image. A local variable called `deviation` was initialized to 0. The position track was then traversed, accessing the `x` and `y` coordinates for each time-stamp. Each set of coordinates were mapped to the corresponding pixel in the landmark pixel distance map that the position track resided in at the particular time-stamp. Then, the pixel value for that pixel was accessed from the distance map. If the pixel value was greater than the variable `deviation`, `deviation` was set to the current pixel value. After the entire position track was traversed, the `deviation` variable held the maximum deviation for that position track.

After deviation values for all position tracks were determined, the feature found the greatest of the 22 maximum deviation values. All 22 maximum deviation values were divided by this value to normalize the deviation scores for all position tracks between zero and one. A segment of MATLAB® code is included to illustrate how this feature algorithm was implemented.

```

deviation= 0;
    for i = 1:length(track)
        x = floor(track(i,2));
        y = floor(track(i,3));
        current_value = roadParkingMap(y,x);
        % x and y reversed flipped because of matrix form
        if current_value > deviation
            deviation = current_value;
        end
    end
end

```

Figure 3.9 presents the landmark distance map for this landmark feature.

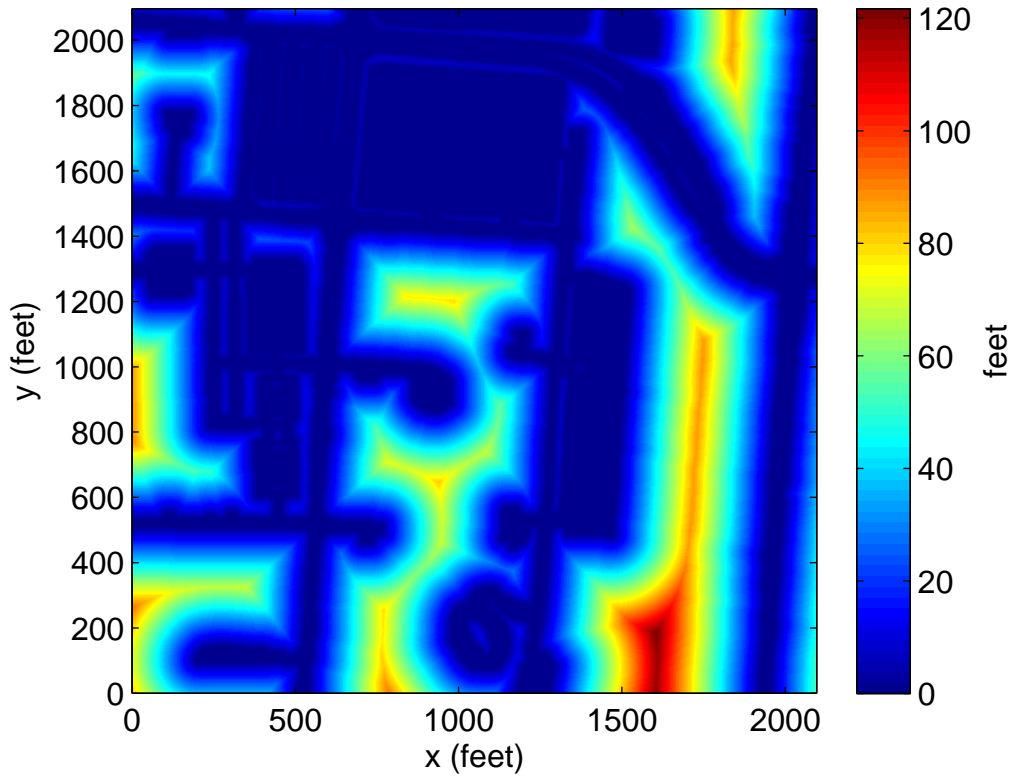


Figure 3.9: Deviation from Roads and Parking Lots Distance Map.

This figure shows the areas of the map that were roads and parking lots (dark blue-colored areas), and also the areas that were not roads and parking lots. The pixels that have significant distance from a road or parking lot were colored appropriately. Each pixel's value in the map represented the distance to the closest parking lot or road.

This subsection described how feature data from the roads and parking lots distance map was generated for the 22 position tracks. The maximum deviation from a road or parking lot within the confines of the overhead image was calculated for each position track. Each deviation value was divided by the maximum deviation of any track in the database to scale all the deviation values from zero to one. The next subsection describes the process used to generate feature data from the proximity to high-valued building landmark feature.

3.3.6 Proximity to High-valued Building

This feature reported the nearest proximity of a position track to a particular building in the overhead image. This building was located just north of the AFIT east parking lot at the Lat-Long coordinates [39.783944N, 84.081043W]. Four satellite dishes were enclosed behind a fence at the south-east corner of the building. The building was selected as a high-valued building for the purpose of this research.

A local variable called `maximumProximity` was initialized to 60. The position track was then traversed, accessing the x and y coordinates for each time-stamp. Each set of coordinates were mapped to the corresponding pixel in the landmark pixel distance map that the position track resided in at that particular time-stamp. Then, the pixel value for that pixel was accessed from the distance map. If the pixel value was less than the proximity variable, proximity was set to the current pixel value. After the entire position track was traversed, the proximity variable held the closest proximity value to the high-valued building for that position track.

After proximity values for all position tracks were determined, the algorithm found the greatest of the 22 maximum deviation values. All 22 maximum deviation values were divided by this value so that the now-scaled values were all real numbers between zero and one. A segment of MATLAB[®] code is included to illustrate how this feature algorithm was implemented.

```

maximumProximity = 60;
    for i = 1:length(track)
        x = floor(track(i,2));
        y = floor(track(i,3));
        current_value = HighValuedBuildingMap(y,x);
        % x and y reversed because of matrix form
        if current_value < maximumProximity
            proximity = current_value;
        end
    end
end

```

Figure 3.10 presents the landmark distance map for this landmark feature.

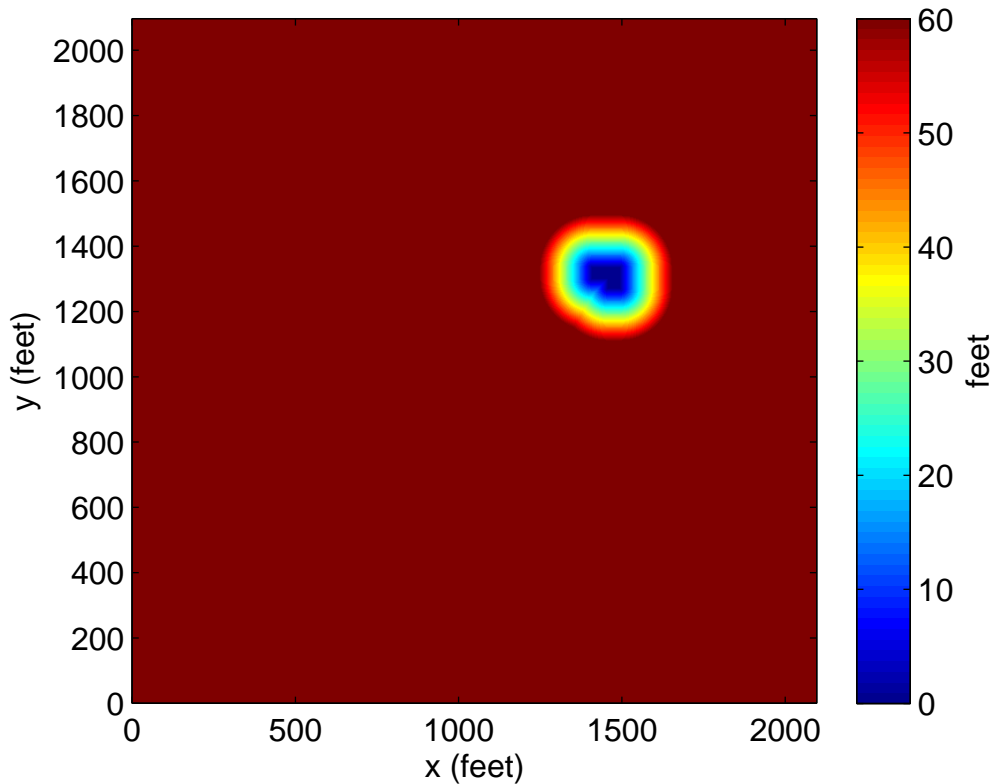


Figure 3.10: Proximity to High-valued Building Distance Map.

This figure shows the area of the map where the high-valued building was located. The maximum value that any pixel in the distance map contained was 60 feet. This subsection described how feature data was generated for the proximity to a high-valued building landmark feature. A score was calculated for each position track of the closest proximity to the high-valued building. The next subsection describes the process used to generate feature data from the proximity to the water tower landmark feature.

3.3.7 *Proximity to Water Tower*

This landmark feature reported the proximity to the water tower on Area B, with the Lat-Long coordinates: [39.784517N, 84.080928W]. This water tower was considered high-

valued for the purpose of this research. Any position track that came within close proximity to it was flagged as suspicious.

A local variable called `maximumProximity` was initialized to 60. The position track was then traversed, accessing the x and y coordinates for each time-stamp. Each set of coordinates were mapped to the corresponding pixel in the landmark pixel distance map that the position track resided in at that particular time-stamp. Then, the pixel value for that pixel was accessed from the distance map. If the pixel value was less than the proximity variable, proximity was set to the current pixel value. After the entire position track was traversed, the proximity variable held the nearest proximity value to the water tower for that position track. A segment of MATLAB® code is included to illustrate how this feature algorithm was implemented.

```
maximumProximity = 60;
    for i = 1:length(track)
        x = floor(track(i,2));
        y = floor(track(i,3));
        current_value = waterTowerMap(y,x);
        % x and y reversed because of matrix form
        if current_value < maximumProximity
            proximity = current_value;
        end
    end
end
```

Figure 3.11 presents the landmark distance map for this landmark feature.

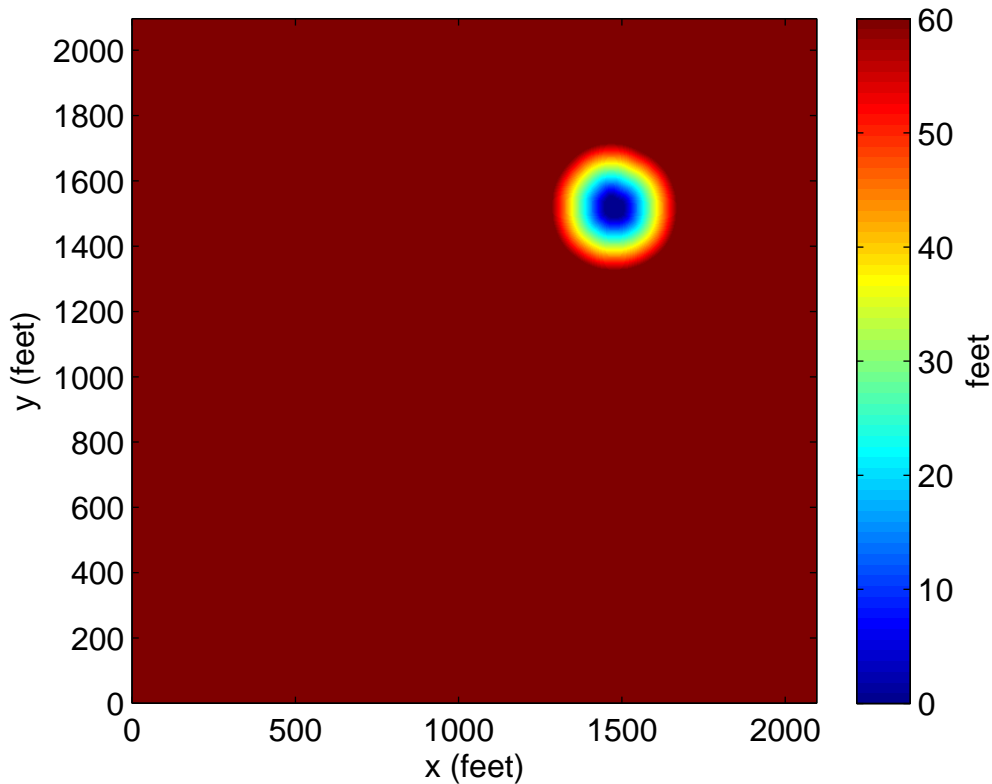


Figure 3.11: Proximity to Water Tower Distance Map.

This figure shows the area of the map where the water tower was located. The maximum value that any pixel in the distance map contained was 60 feet. This subsection described how feature data was generated for the proximity to water tower feature. A score was calculated for each position track of the closest proximity to the water tower.

This section on described the processes employed to generate feature data from the position track database. The first subsection described the development of the grid necessary to calculate data using the dwell and repetition features. Then, the dwell and repetition feature algorithms were explained. The development of the landmark distance maps was then discussed. Finally, the three landmark features were described. The next section describes how each position track was classified using the LOOCV method.

3.4 Classification of Position Tracks

This section details the process of individually classifying each collected position track in the position track database using the LOOCV method. Each position track was classified using the four classification methods introduced in chapter II. Before classification could be performed, the feature generation data had to be converted into three matrices for the `classify` function.

3.4.1 Pre-classification Data Processing

Before a position track could be classified as suspicious or non-suspicious, the generated feature data needed to be processed into a format acceptable for the `classify` function. First, the generated feature data was consolidated into a single matrix in MATLAB®. This matrix had the dimensions of 22 x 5 (22 tracks and five features). Next, this matrix was segregated into two matrices for each class (suspicious and non-suspicious). The suspicious class matrix had the dimensions of 12 x 6 while the non-suspicious class matrix had dimensions of 10 x 6. The additional column in these matrices stored the track number of each position track for ease of error checking.

The two matrices grouped by class were then used to create the sample, training set and group matrices required by MATLAB®'s `classify` function. The sample matrix consisted of the particular position track to be classified as suspicious or non-suspicious by the `classify` function. It was essentially a vector that consisted of the feature data generated for the track to be classified. For this research, the dimension of the sample matrix was always 1 x 5. While the `classify` function was able to classify more than one position track simulatenously, position tracks were always classified one a time in this research because the LOOCV method was employed.

The training data matrix consisted of the generated feature data for the other 21 tracks that were not being classified. In this research, the dimensions of this matrix were always 21 x 5. The group matrix always had dimensions of 21 x 1 and was essentially a grouping

vector for the training data matrix. It instructed the classify function which position tracks in the training data matrix were pre-determined to be suspicious or non-suspicious.

The value in a particular row of the group matrix indicated the class of the track corresponding to the same row in the training data matrix. If a certain position track was suspicious, a zero was assigned to that track's corresponding row in the group matrix. If a position track was designed to be non-suspicious, the value in the group vector was set to one. For example, if a certain track in the fourth row of the training data matrix was designed to be a suspicious track, the fourth row of the group matrix was set to zero.

Code was written in MATLAB® to consolidate the generated feature data for the 22 position tracks into one matrix, create the two matrices separated by class and finally populate the sample, training data and group matrices required for the classify function. Figure 3.12 illustrates example sample, training data and group matrices used in this research. The first value in the sample matrix indicates track nine was the position track that was classified. The first column in the training data matrix displays the position track numbers. This column was inserted for error checking and was not passed into the classify function. Columns two through six in this matrix display the generated feature data for the position tracks in the following order: dwell, repetition, deviation from roads and parking lots, proximity to the high-valued building and proximity to the water tower.

```

Command Window
>> sample
sample =
    9.0000    0.4570    0.4737    0.8249    0.5311    0.4386
>> training_data
training_data =
    1.0000    0.3775    0.3421    0.4076    1.0000    0.8638
    2.0000    1.0000    0.1579    0.2762         0         0
    3.0000    0.3775    0.1579    0.8654    0.3238    0.2266
    8.0000    0.7417    0.5789    0.2113    0.6054    0.6241
   10.0000    0.2318    0.0789    0.5902    0.8809    1.0000
   12.0000    0.3179    0.0789    0.5339    0.9068    0.7689
   15.0000    0.6291    0.2632    0.5836    0.5880    0.6117
   17.0000    0.5033    0.1053    0.8750    0.5312    0.4377
   19.0000    0.2318    1.0000    0.0766    0.5987    0.4848
   20.0000    0.5497    0.1053    0.7796    0.2897    0.2216
   22.0000    0.3510    0.1316    1.0000    0.3625    0.2624
    4.0000    0.3709    0.0263    0.5327    0.6205    0.6419
    5.0000    0.4901         0    0.8560    0.4093    0.3161
    6.0000    0.0662    0.0263    0.2762    0.5924    0.6154
    7.0000    0.2053         0    0.8560    0.5047    0.3946
   11.0000    0.0795         0    0.8249    0.5858    0.6082
   13.0000    0.3841         0    0.8699    0.6734    0.5721
   14.0000    0.3444    0.0263    0.4358    0.6336    0.6522
   16.0000    0.5166         0    0.8767    0.5341    0.4404
   18.0000    0.3510         0    0.8767    0.5379    0.4440
   21.0000    0.3709    0.0263    0.5507    0.5388    0.4450
>> group'
ans =
Columns 1 through 15
    0     0     0     0     0     0     0     0     0     0     0     1     1     1     1
Columns 16 through 21
    1     1     1     1     1     1
fx >>

```

Figure 3.12: Example of Sample, Training Data and Group Matrices in MATLAB®.

After the sample, training set and group matrices were generated in MATLAB®, classification of a single position track using the four different classification methods could then be accomplished.

3.4.2 Classification in MATLAB®

Position tracks were classified as suspicious or non-suspicious with MATLAB®'s `classify` function using the sample, training set and group matrices described in subsection 3.4.1. Each position track in the database was classified four times using the LOOCV method by the four classification methods discussed in chapter II. The `classify`

function always returned either a zero or one for each of the four classification methods since the intent classifier in this research was a two class scenario.

If a zero was returned, the position track in the sample matrix was classified as suspicious. Conversely, the position track in the sample matrix was classified as non-suspicious if a one was returned. After a position track was classified as suspicious or non-suspicious, the results were stored for statistical processing of the four classification methods. Chapter IV will discuss the results of the four classification methods.

This section explained the process performed to individually classify each collection position track in the position track database using the LOOCV method. Each position track was classified using the four classification methods introduced in chapter II. Subsection 3.4.1 explained how the generated feature data was organized into the sample, training data and group matrices needed for classification. Subsection 3.4.2 communicated how classification of the position tracks was performed in MATLAB®.

3.5 Chapter Summary

This chapter described the processes and methodologies that were performed to collect and process x-y position tracks using the Magellan® Mobilemapper GPS unit, generate features from the collected position tracks and then process the generated feature data to perform intent classification on a single position track in MATLAB®.

Chapter IV will discuss the feature generation and classification results produced in this research using the methodologies that were described in this chapter.

IV. Results and Analysis

This chapter discusses the feature generation and classification results produced in this research using the methodologies described in chapter III. A detailed analysis of these results is performed. Each position track in the database was individually classified using the LOOCV method by the four classification methods discussed in chapter II. Cross-sectional feature generation data plots that include discriminant lines and curves from the four classification methods are presented and discussed in section 4.1. The classifier statistics and results are presented and analyzed in section 4.2.

4.1 Feature Generation Results

Cross-sectional feature generation figures are presented and explained in this section. Figure 4.1 shows a cross-sectional plot of the dwell time vs. repetition feature generated feature data. 162 foot grid cell spacing was chosen for this figure because it allowed for the greatest separation between suspicious and non-suspicious tracks. The four discriminant function lines and curves from the four classification methods discussed in chapter II are also shown.

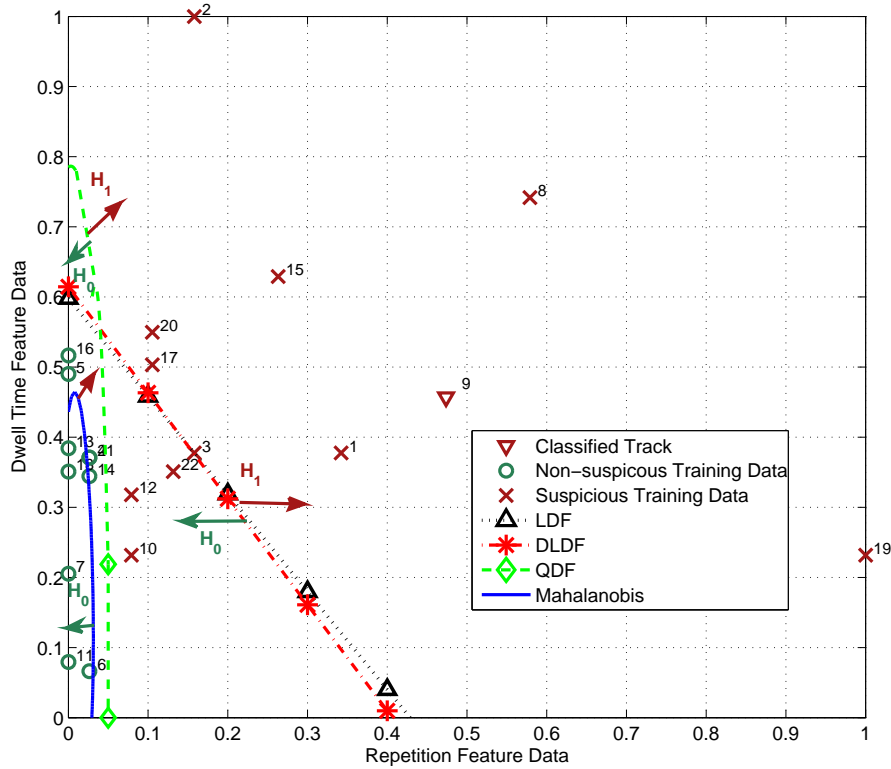


Figure 4.1: Dwell Time vs. Repetition: $w_{cell} = 162$ feet, Classified Track = 9.

The red upside-down triangle was the position track that was classified. Track nine was the position track that was classified for all of the two dimensional cross-sectional plots in this section. The green circles and red x's represent the non-suspicious and suspicious position tracks used as training data for the classifier, respectively. The discriminant function lines and curves display the decision boundary areas for each classification method. A boundary area labeled as H_0 indicates the area that the particular classification method would classify a position track as non-suspicious. Likewise, a boundary area labeled as H_1 indicates the area that the particular classification method would classify a position track as suspicious.

The position and shape of the discriminant function lines and curves depended on the track that was classified as suspicious or non-suspicious because the training data used as an input into the classifier would change. Feature data from only the dwell time and repetition features were used by the classifier in Figure 4.1. All four methods correctly classified track nine as suspicious. $P[H_0]$ and $P[H_1]$ were both set to 0.5 in MATLAB® when the `classify` function was executed. All position tracks that had a repetition score of about 0.1 and greater were suspicious, while some non-suspicious position tracks had a dwell time score as high as 0.5.

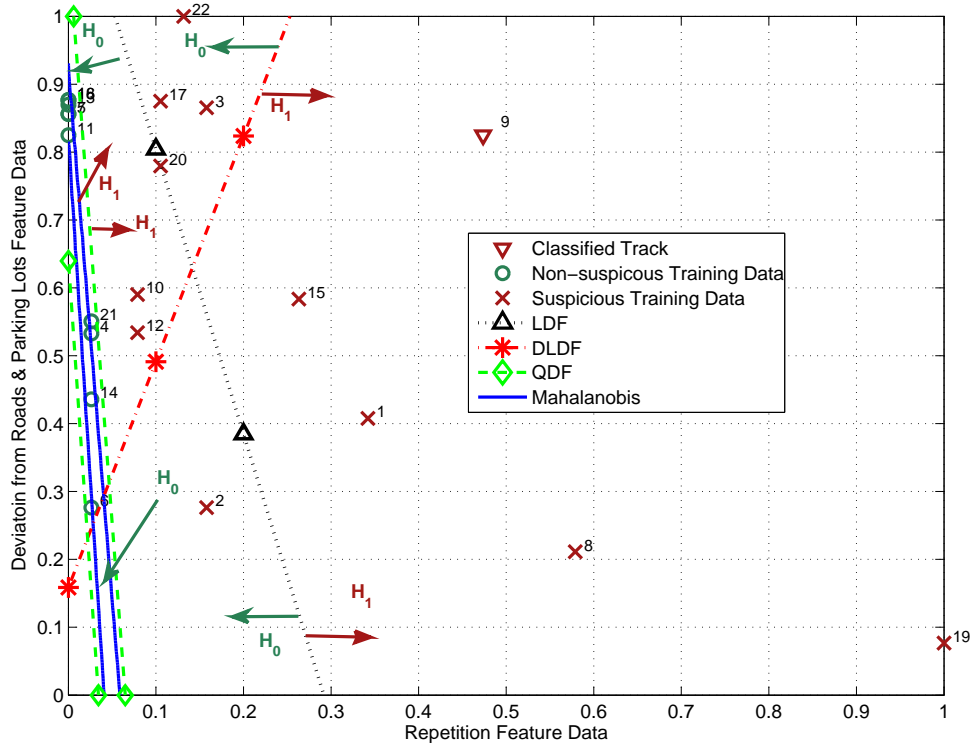


Figure 4.2: Deviation from Roads and Parking Lots vs. Repetition: $w_{cell} = 162$ feet, Classified Track = 9.

Figure 4.2 shows a cross-sectional plot of the deviation from roads and parking lots vs. repetition generated feature data. Again, the red upside-down triangle represented track nine and was the position track that was classified. Feature data from only the deviation from roads and parking lots and repetition features were used by the classifier. All four classification methods correctly classified track nine as suspicious. $P[H_0]$ and $P[H_1]$ were both set to 0.5 in MATLAB® when the `classify` function was executed. 162 foot grid cell spacing was again chosen for this figure because it allowed for the greatest separation between suspicious and non-suspicious tracks. While all position tracks that had a repetition score of about 0.1 and greater were suspicious, some non-suspicious position tracks had a deviation from roads and parking lots score as high as 0.9.

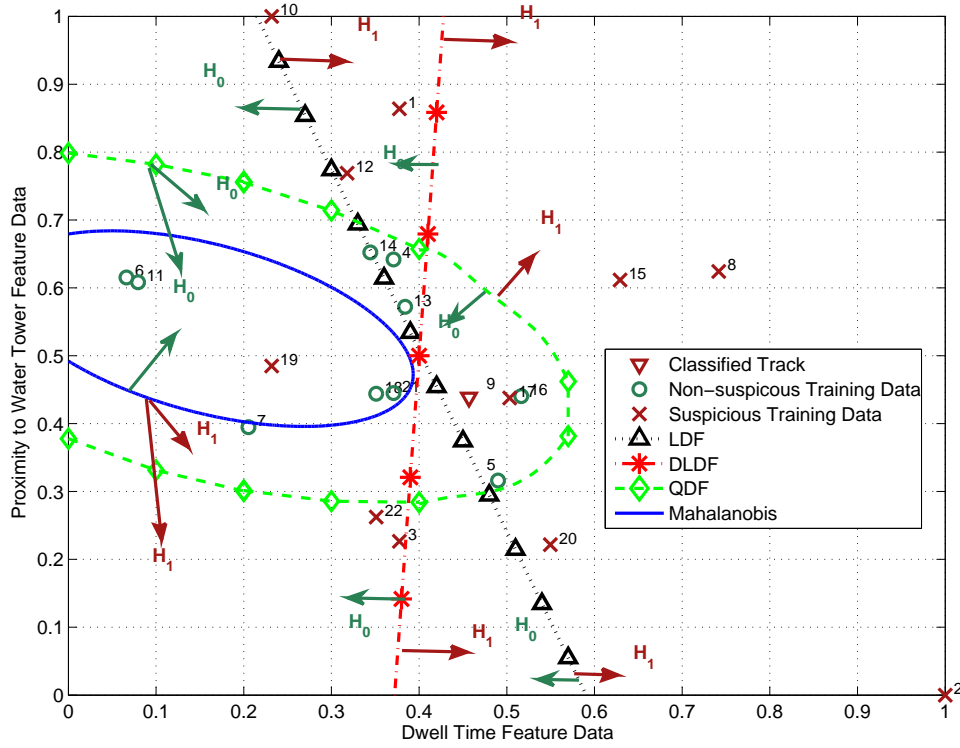


Figure 4.3: Proximity to Water Tower vs. Dwell Time: $w_{cell} = 162$ feet, Classified Track = 9.

Figure 4.3 shows a cross-sectional plot of the proximity to the water tower vs. dwell time generated feature data. Track nine was the position track that was classified. Feature data from only the dwell time and proximity to water tower were used by the classifier. The LDF, DLDF and Mahalanobis methods correctly classified track nine as suspicious, while the QDF method incorrectly classified it as non-suspicious.

One possible reason why the QDF curve was pulled to the right in this figure was because track 16 had a dwell score of 0.5 but was collected with the intent of being non-suspicious. If this position track were not included in the training data, the QDF method may have correctly classified track nine as suspicious. 162 foot grid cell spacing was again

chosen for this figure because it allowed for the greatest separation between suspicious and non-suspicious tracks.

Figure 4.4 shows a cross-sectional plot of the proximity to the high-valued building vs. repetition generated feature data. Track nine was the position track that was classified. Feature data from only the repetition and proximity to high-valued building features were used by the classifier. All four classification methods correctly classified track nine as suspicious. 162 foot grid cell spacing was again chosen for this figure because it allowed for the greatest separation between suspicious and non-suspicious tracks. While all position tracks that had a repetition score of about 0.1 and greater were suspicious, some non-suspicious position tracks had a proximity to the high-valued building score as high as almost 0.7.

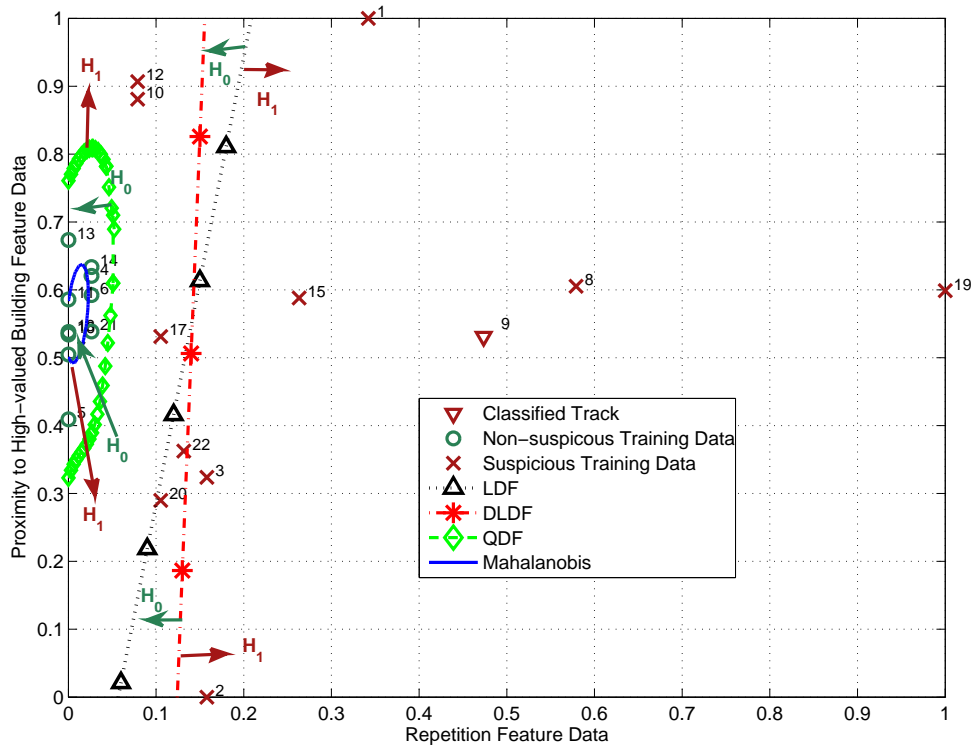


Figure 4.4: Proximity to High-Valued Building vs. Repetition: $w_{cell} = 162$ feet, Classified Track = 9.

Four cross-sectional feature generation data plots that included discriminant function lines and curves from the four different classification methods were presented and discussed in this section. These figures verified the functionality of the five feature generation algorithms. Additionally, it was shown that the discriminant function lines and curves for the four classification methods varied appropriately depending on which position track in the database was classified. The next section presents the classifier results and statistics.

4.2 Classifier Results

This section communicates the quantitative results of the classifier employed in this intent assessment system. Subsection 4.2.1 presents and analyzes the results generated

from a sweep of the grid resolution parameter, w_{cell} . In subsection 4.2.2, results from a sweep of the prior probability of the suspicious class ($P[H_1]$) are presented and discussed. Finally, subsection 4.2.3 presents P_D vs. P_F data for each classification method and the corresponding best fit ROC curves.

4.2.1 Grid Resolution Sweep Results

In this subsection, a sweep of the grid cell dimension parameter w_{cell} (in feet) was performed on the classifier and the results are presented and analyzed. This parameter changed the resolution of the grid which affected the data generated by the dwell and repetition features.

The classifier performance and behavior varied as w_{cell} changed because the dwell and repetition feature generation data depended on the grid resolution. The grid cell spacing parameter w_{cell} was incremented from 2.7 feet (or 1 image pixel unit) to 1,026 feet (380 image pixel units). The maximum value for w_{cell} did not exceed 1,026 feet because at that grid resolution there were only four grid cells. If w_{cell} were set to a longer cell dimension, the entire 2,095 feet x 2,095 feet grid would have been only one cell. The dwell and repetition algorithms would have failed to produce accurate feature data.

The parameter sweep comprised of 22 different w_{cell} values. The classifier classified each position track in the database for each different w_{cell} value, using the four classification methods. The prior probability values $P[H_0]$ and $P[H_1]$ for the non-suspicious and suspicious classes respectively were both set to 0.5 in MATLAB® for the entire sweep. Figure 4.5 illustrates the overall accuracy of the classifier for the grid resolution sweep.

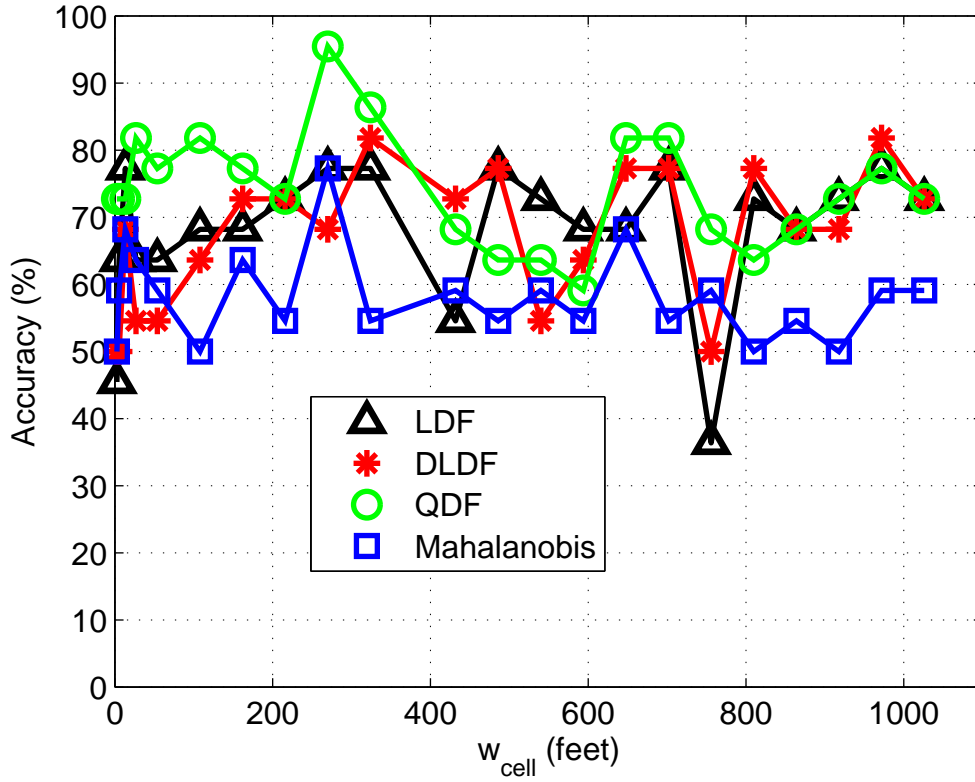


Figure 4.5: Overall Classifier Accuracy for Grid Resolution Sweep.

On average, the QDF classification method outperformed the other three classification methods in terms of overall accuracy. The definition of accuracy in the context of this research was the rate at which the classifier correctly classified a position track from the database as non-suspicious or suspicious, using the LOOCV method. The average performance for the QDF method over the entire grid resolution spectrum was 74.14%.

The Mahalanobis classification method performed the worst of the four classification methods. The average performance of the Mahalanobis method for the grid resolution sweep was 58.26%. While the Mahalanobis classification method had the worst average accuracy, it still performed (only slightly) better than the random guess accuracy rate (50%).

A classification method that randomly classifies a position track as suspicious or non-suspicious is accurate about 50% of the time. Therefore, a classification method with a defined algorithm for classifying data that has an accurate rate of less than 50% is no better than the random guess method.

The best accuracy for the entire plot was performed by the QDF classification method when w_{cell} was 270 feet. The QDF accuracy for that w_{cell} value was 95.45%. The worst accuracy displayed in the plot was performed by the LDF classification method when w_{cell} was 756 feet. The LDF accuracy for that w_{cell} value was 36.36%. If a military installation were picking a classification method solely based on maximizing the classifier accuracy, the QDF method should be selected, with w_{cell} set to 270 feet.

Tables 4.1 and 4.2 present confusion matrices using all five features for the four classification methods for the grid resolution sweep. These matrices present the classifier statistics when w_{cell} was set to 270 feet. The rows of the confusion matrices indicate the actual classes that the position tracks belonged to and the columns indicate the classes that the classification methods assigned each position track as.

| | LDF | | DLDF | |
|----------------|----------------|------------|----------------|------------|
| | Non-suspicious | Suspicious | Non-suspicious | Suspicious |
| Non-suspicious | 8 | 2 | 7 | 3 |
| Suspicious | 3 | 9 | 4 | 8 |

Table 4.1: Confusion Matrix for LDF and DLDF Methods: $w_{cell} = 270$ feet.

The matrices display the number of position tracks in the database that were correctly classified into each class, and the number of tracks that were incorrectly classified into each class. The LDF classification method correctly classified eight non-suspicious tracks in the database as non-suspicious. Two non-suspicious tracks were incorrectly classified as

suspicious. Three suspicious tracks were incorrectly classified as non-suspicious and nine suspicious tracks were correctly classified as suspicious.

The DLDF classification method correctly classified seven non-suspicious tracks in the database as non-suspicious. Three non-suspicious tracks were incorrectly classified as suspicious. Four suspicious tracks were incorrectly classified as non-suspicious and eight suspicious tracks were correctly classified as suspicious.

| | QDF | | Mahalanobis | |
|----------------|----------------|------------|----------------|------------|
| | Non-suspicious | Suspicious | Non-suspicious | Suspicious |
| Non-suspicious | 9 | 1 | 5 | 5 |
| Suspicious | 0 | 12 | 0 | 12 |

Table 4.2: Confusion Matrix for QDF and Mahalanobis Methods: $w_{cell} = 270$ feet.

The QDF classification method correctly classified nine non-suspicious tracks in the database as non-suspicious. One non-suspicious track was incorrectly classified as suspicious. No suspicious tracks were incorrectly classified as non-suspicious and all 12 suspicious tracks were correctly classified as suspicious.

The Mahalanobis classification method correctly classified five non-suspicious tracks as non-suspicious. The other five non-suspicious tracks were incorrectly classified as suspicious. No suspicious tracks were incorrectly classified as non-suspicious and all 12 suspicious tracks were correctly classified as suspicious.

The false negative error statistic P_M vs. w_{cell} and the false positive error statistic P_F vs. w_{cell} performances are now presented for the grid resolution sweep. Figure 4.6 presents the false negative error statistic P_M of the classifier as function of w_{cell} .

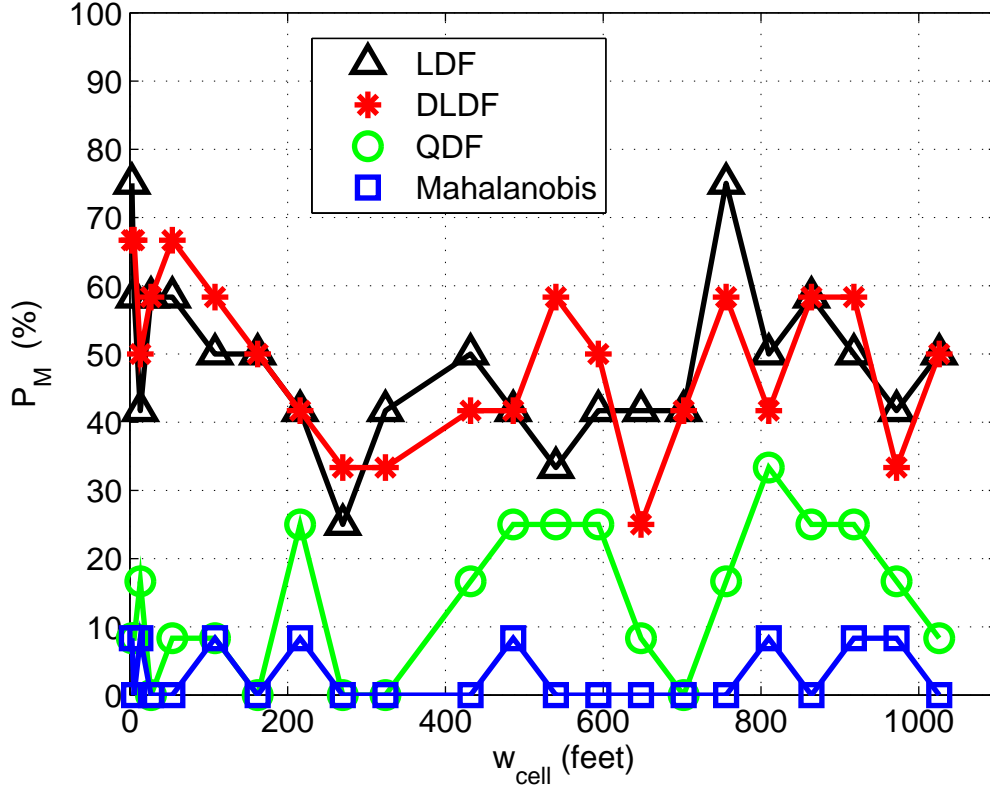


Figure 4.6: P_M Performance for Grid Resolution Sweep.

P_M was defined mathematically as $P[H_0|H_1]$ (H_0 picked when H_1 was true) and was the ratio of position tracks incorrectly classified as non-suspicious compared to the total number of suspicious tracks in the database. It was the likelihood that the classifier missed a suspicious track and classified it as non-suspicious. This metric was potentially more important to security personnel than the overall classifier accuracy because the consequences of mis-classifying a suspicious position track as non-suspicious were severe. A hostile intruder could comprise an installation's security without its activity being flagged as suspicious.

On average, the Mahalanobis classification had the lowest P_M rate: 3.03%. The LDF and DLDF classification methods had the worst P_M rates at 48.86% and 49.24%

respectively. The Mahalanobis classification method yielded a P_M of 0% 14 out of the 22 different w_{cell} values, or more often than not. The QDF classification method had an average P_M of 13.64%. This classification method also yielded a P_M of 0% five out of the 22 different w_{cell} values. When w_{cell} was set to 270 feet, the QDF classification method had a P_M of 0%.

If a military installation were choosing a single classification method solely based on minimizing P_M , the Mahalanobis method should be selected. Figure 4.7 presents the false positive error rate P_F of the classifier as a function of w_{cell} .

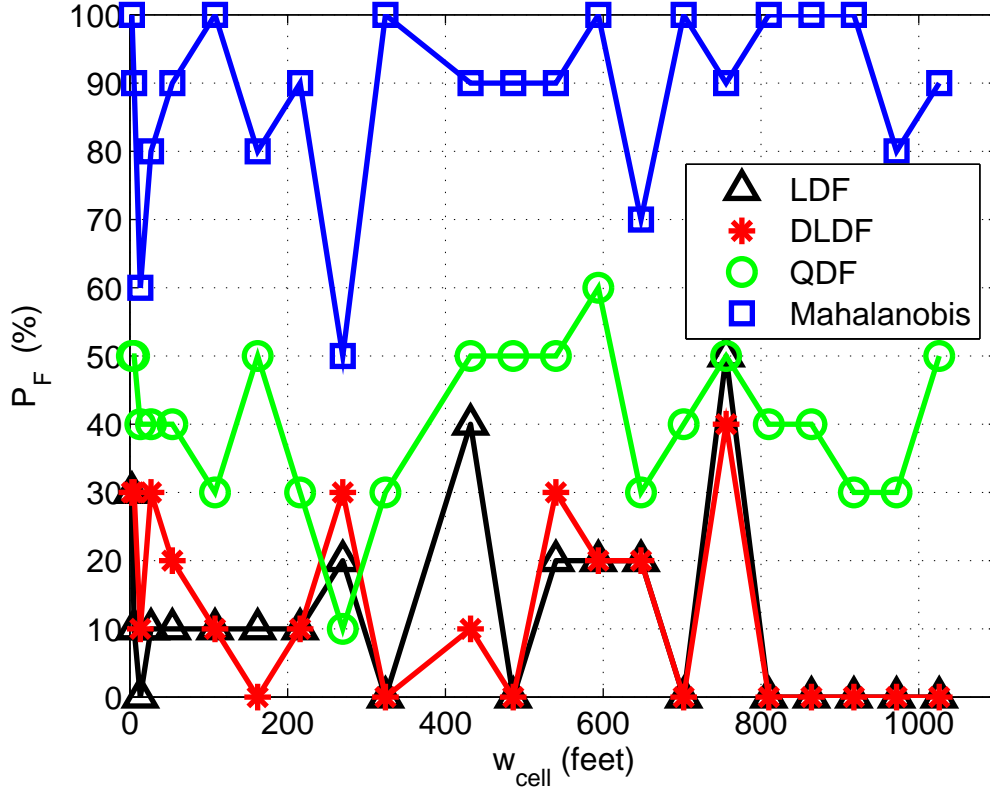


Figure 4.7: P_F Classifier Performance for Grid Resolution Sweep.

P_F was defined mathematically as $P[H_1|H_0]$ (H_1 picked when H_0 was true) and was the ratio of position tracks incorrectly classified as suspicious compared to the total number of non-suspicious tracks in the database. It was the likelihood that the classifier incorrectly classified a non-suspicious track as suspicious. The consequences of mis-classifying a non-suspicious position track as suspicious were not as severe as incorrectly classifying a suspicious track as non-suspicious.

In the event that a non-suspicious position track was classified as suspicious, security personnel would be dispatched to investigate the emitting object. The act of dispatching personnel to investigate the emitter would expend resources in various forms such as man-hours. The cost of missing a suspicious track and subsequently not engaging it as described

earlier in this subsection outweighs the cost of expending these resources to investigate the uncertain intent of an emitter that was non-suspicious.

On average, the LDF classification method had the lowest P_F rate: 11.82%. The DLDF classification method had the second lowest P_F rate: 13.18%. The QDF classification method had the worst P_F rate: 40.45%. The LDF classification method yielded a P_F of 0% nine out of the 22 different w_{cell} values. This classification method had a P_F of 0% for 41% of the resolution sweep. When w_{cell} was set to 270 feet, the QDF classification method had a P_F of 10.00%. 10.00% was the lowest P_F probability that the QDF method achieved for the grid resolution sweep.

If a military installation were picking a single classification method solely based on minimizing P_F , the LDF method should be selected. The quantitative results for the grid resolution sweep are summarized in Table 4.3.

| Statistics | LDF | DLDF | QDF | Mahalanobis |
|------------------------------------|-------|-------|-------|-------------|
| Average Accuracy (%) | 67.98 | 67.15 | 74.17 | 58.26 |
| Standard Deviation of Accuracy (%) | 10.63 | 10.58 | 8.58 | 6.82 |
| Average P_M (%) | 48.86 | 49.24 | 13.64 | 3.03 |
| P_M Standard Deviation (%) | 11.87 | 12.04 | 10.46 | 4.10 |
| Average P_F (%) | 11.82 | 13.18 | 40.45 | 88.18 |
| P_F Standard Deviation (%) | 14.02 | 13.59 | 11.33 | 13.68 |

Table 4.3: Summary of Classifier Statistics for Grid Resolution Sweep.

This subsection communicated the results of the grid resolution sweep. The grid cell spacing parameter w_{cell} was incremented from 2.7 feet (or 1 image pixel unit) to 1,026 feet (380 image pixel units). The overall classifier accuracy, P_M and P_F were displayed as a function of w_{cell} . The figures were analyzed and recommendations were made concerning which classification method was the most appropriate in terms of the three statistics. The QDF classification method had the best overall accuracy of the four classification methods. The Mahalanobis classification method had the lowest average P_M rate and the LDF method had the lowest average P_F rate. In the next subsection, the results generated from parameter sweeps of the suspicious class prior probability $P[H_1]$ are presented and analyzed.

4.2.2 Prior Probability Sweep Results

In this subsection, a second parameter sweep was performed and the results were observed. The prior probability of the suspicious class $P[H_1]$ was incremented from 0.5 to 0.9 in 0.05 increments while the grid cell spacing parameter w_{cell} was kept constant. The $P[H_1]$ value instructed the classifier of the likelihood that a position track was suspicious. The $P[H_1]$ sweep started at 0.5 and was increased from that value because it was not appropriate in the context of the intent assessment in this research to have a $P[H_1]$ value less than 0.5. A $P[H_1]$ value less than 0.5 meant that the likelihood of a suspicious track occurring was less than 50%, or half the time.

If the classifier was instructed that the likelihood of a suspicious track occurring was less than 50%, it was more likely to classify a track as non-suspicious. Consequently, the P_M rate would increase which would have adverse effects for a military installation (as discussed in subsection 4.2.1). As $P[H_1]$ was incremented from 0.5 to 0.9, $P[H_0]$ was decremented from 0.5 to 0.1. The sum of $P[H_1]$ and $P[H_0]$ always added to one. $P[H_0]$ was the prior probability of the non-suspicious class and likewise instructed the classifier of the likelihood that a position track was non-suspicious.

The $P[H_1]$ sweep was performed to observe the classifier's overall accuracy, P_M and P_F probabilities as $P[H_1]$ increased. As $P[H_1]$ increased, it was expected that P_M would decrease and P_F would increase. P_M would decrease as $P[H_1]$ increases because the classifier would be more sensitive to suspicious tracks and was more likely to classify a track as suspicious. The number of suspicious position tracks incorrectly classified as non-suspicious would decrease. Likewise, P_F would increase as $P[H_1]$ was increased because the number of non-suspicious tracks classified as suspicious would increase. The classifier was more sensitive to suspicious tracks and was more likely to classify a track as suspicious.

The classifier classified each position track in the database for each different $P[H_1]$ value using the four classification methods. The results of two $P[H_1]$ sweeps conducted for two different w_{cell} values are presented here. The w_{cell} values chosen for this subsection's results were 270 and 918 feet. w_{cell} was set to 270 feet for the first $P[H_1]$ sweep because the highest accuracy for the QDF classification method (95.45%) occurred at this grid cell width value. Therefore, a $P[H_1]$ sweep with this w_{cell} value would theoretically yield the best classifier results. w_{cell} was set to 918 feet for the second $P[H_1]$ sweep so the first sweep results were compared to a $P[H_1]$ sweep with w_{cell} set to a value in the top tenth of the grid resolution spectrum.

The classifier's overall accuracy, P_M and P_F rates vs. $P[H_1]$ are presented for each $P[H_1]$ sweep. Figure 4.8 presents the overall accuracy of the classifier for a sweep of $P[H_1]$ when w_{cell} was set to 270 feet.

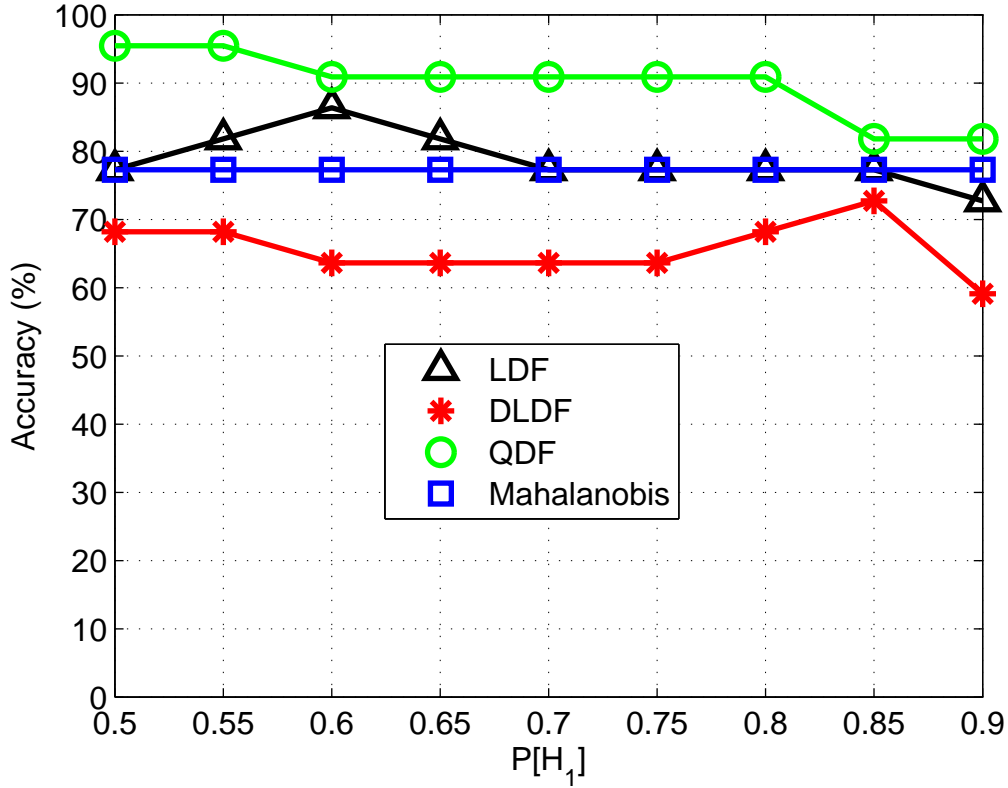


Figure 4.8: Classifier Accuracy for $P[H_1]$ Sweep: $w_{cell} = 270$ feet.

The QDF classification method was the most accurate of the four classification methods for the entire sweep of $P[H_1]$ when w_{cell} was set to 270 feet. The average performance for the QDF method for this sweep was 89.90%. As $P[H_1]$ increased, the accuracy of the QDF classification method gradually decreased, and never increased.

The best accuracy for the QDF method was 95.45% and occurred when $P[H_1]$ was 0.5 and 0.55. The QDF's accuracy standard deviation for the $P[H_1]$ sweep was 4.97%. This method's accuracy varied less for this sweep when compared to the method's accuracy for the grid resolution sweep that had a standard deviation of 8.58%.

The accuracy for the Mahalanobis method remained constant at 77.27%. The accuracy for the LDF classification method increased slightly at first and then decreased slightly as

$P[H_1]$ increased. Finally, the DLDF classification method's accuracy decreased slightly when $P[H_1]$ was 0.6 and then increased to 72.76% when $P[H_1]$ was 0.85. All four classification method's accuracies varied less in this $P[H_1]$ sweep when w_{cell} was set to 270 feet compared to the classification method's accuracies for the grid resolution sweep.

Figure 4.9 presents the false negative error rate P_M of the classifier as a function of $P[H_1]$ for the four classification methods. w_{cell} was set to 270 feet.

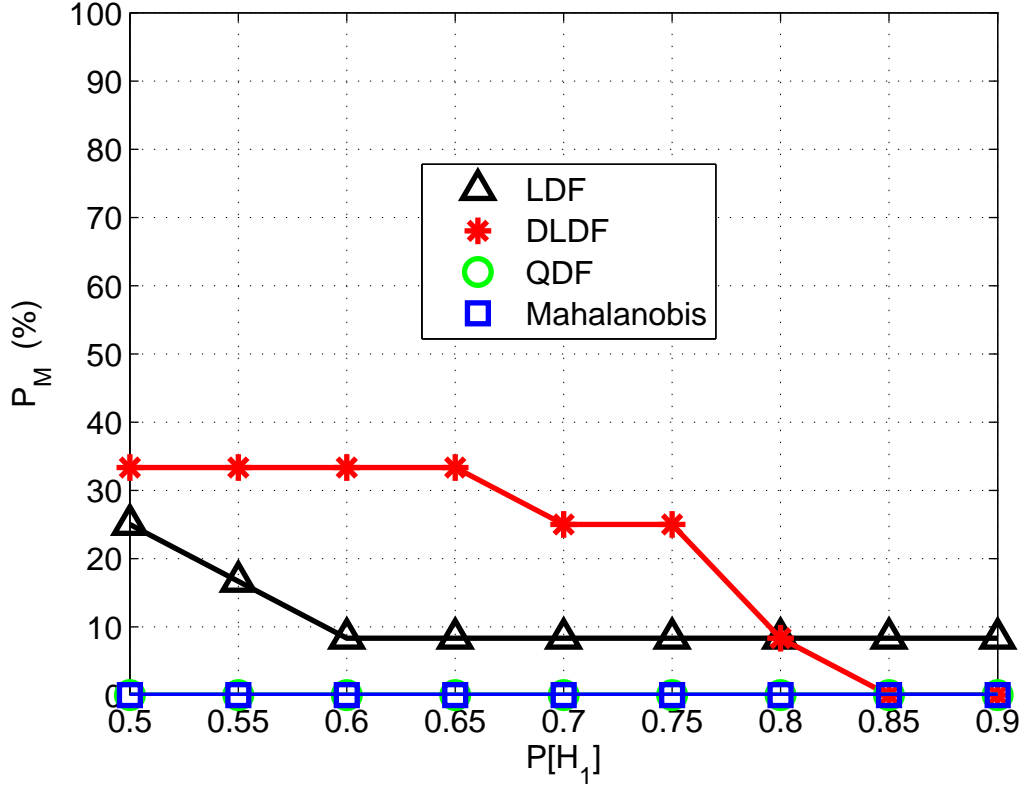


Figure 4.9: P_M Performance for $P[H_1]$ Sweep: $w_{cell} = 270$ feet.

The QDF and Mahalanobis classification methods had P_M values of 0% for the entire sweep. This means that there were no false negative errors for these two methods. The classification methods correctly classified all suspicious tracks as suspicious. This was ideal because as stated in subsection 4.2.1, the consequences of mis-classifying a suspicious position track as non-suspicious were severe.

The DLDF classification method started with a P_M value of 33% when $P[H_1]$ was 0.5. As $P[H_1]$ increased, P_M for this method decreased, leveled once at 25% and then decreased to 0% for $P[H_1]$ values of 0.85 and 0.9. Finally, the LDF classification method had a P_M value of 25% at the start of the sweep when $P[H_1]$ was 0.5. As $P[H_1]$ increased, the P_M for

this method decreased linearly to 8.3% when $P[H_1]$ was 0.6, and remained at 8.3% for the rest of the sweep.

The standard deviation of the P_M values for the LDF, QDF and Mahalanobis classification methods for this $P[H_1]$ sweep were less than the standard deviation of the P_M values for the same classification methods for the grid resolution sweep. The DLDF classification method's P_M standard deviation for the $P[H_1]$ sweep was greater than the method's corresponding standard deviation for the grid resolution sweep.

It was expected that as $P[H_1]$ approached 0.9 during the sweep that the P_M rates for all classification methods would approach 0%. As $P[H_1]$ increased, the classifier became more sensitive to suspicious tracks and was more likely to classify a position track as suspicious. It was shown in this figure that the number of false negative errors decreased for the DLDF and LDF classification methods as $P[H_1]$ increased. As stated earlier, the QDF and Mahalanobis classification methods both had P_M rates of 0% during the entire sweep of $P[H_1]$. The LDF method never achieved a P_M of 0%. The QDF and Mahalanobis classification methods had the best overall P_M rates (0%) for the $P[H_1]$ sweep when w_{cell} was 270 feet.

Figure 4.10 illustrates the false positive error rates P_F of the four classification methods for the $P[H_1]$ sweep. w_{cell} was set to 270 feet.

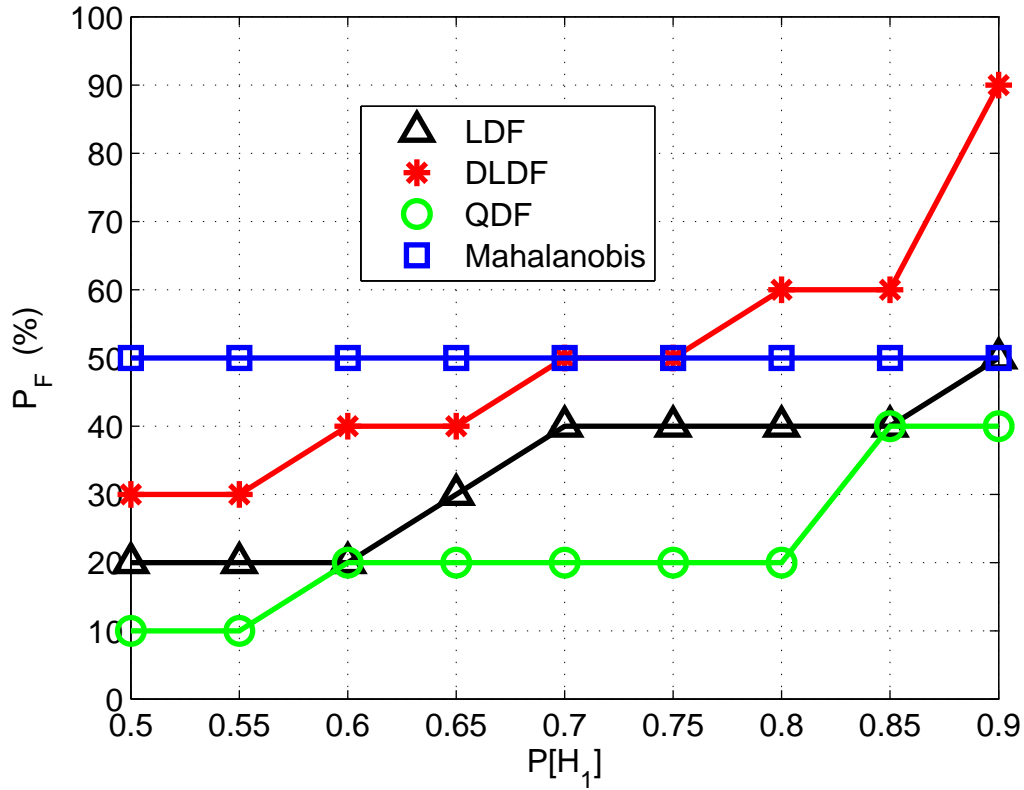


Figure 4.10: P_F Performance for $P[H_1]$ Sweep: $w_{cell} = 270$ feet.

The QDF classification method had the lowest average P_F value: 22.22%. The method had a P_F value of 10% at the beginning of the sweep. The P_F rate then increased to 20% and leveled until $P[H_1]$ was 0.8. The P_F value then increased to 40% at the end of the sweep. The ideal $P[H_1]$ value for this method was 0.5. When $P[H_1]$ was 0.5, the QDF method had no false negative errors (Figure 4.9), and only 10% false positive errors.

The LDF classification method had the second-best average P_F rate: 33.33%. The method had a P_F value of 20% at the beginning of the sweep. The P_F rate then increased linearly to 40% and leveled until $P[H_1]$ was 0.85. The P_F value then increased to 50% at the end of the sweep. The ideal $P[H_1]$ value for this method was 0.6. When $P[H_1]$ was

0.6, the LDF method incorrectly classified 10% of the suspicious tracks as non-suspicious (Figure 4.9), and incorrectly classified 20% of the non-suspicious tracks as suspicious.

The Mahalanobis and DLDF classification methods had the worst average P_F rates: 50%. The Mahalanobis method remained at 50% for the entire sweep. It consistently classified half of the non-suspicious tracks in the database as suspicious during the sweep. The DLDF method had P_F value of 30% at the start of the sweep. As $P[H_1]$ increased, P_F for this method gradually climbed in a stair-step manner until $P[H_1]$ was 0.85. The P_F value then increased from 60% to 90% when $P[H_1]$ increased from 0.85 to 0.9.

It was expected that the P_F rates for all classification methods would increase as $P[H_1]$ approached 0.9 during the sweep. As $P[H_1]$ increased, the classifier became more sensitive to suspicious tracks and was more likely to classify any position track as suspicious. It was shown in Figure 4.10 that the number of false positive errors increased for all of the classification methods as $P[H_1]$ increased except the Mahalanobis method. The quantitative results for the this $P[H_1]$ sweep with w_{cell} set to 270 feet are summarized in Table 4.4.

| Statistics | LDF | DLDF | QDF | Mahalanobis |
|------------------------------------|-------|-------|-------|-------------|
| Average Accuracy (%) | 78.79 | 65.66 | 89.90 | 77.27 |
| Standard Deviation of Accuracy (%) | 3.94 | 4.01 | 4.97 | 0 |
| Average P_M (%) | 11.11 | 21.30 | 0 | 0 |
| P_M Standard Deviation (%) | 5.89 | 14.50 | 0 | 0 |
| Average P_F (%) | 33.33 | 50.00 | 22.22 | 50.00 |
| P_F Standard Deviation (%) | 11.18 | 18.71 | 10.93 | 0 |

Table 4.4: Classifier Statistics for $P[H_1]$ Sweep: $w_{cell} = 270$ feet.

A $P[H_1]$ sweep was conducted with w_{cell} set to 270 feet. The overall accuracy, P_M and P_F rates vs. $P[H_1]$ were presented and analyzed for this sweep. A second $P[H_1]$ sweep was

conducted with w_{cell} set to 918 feet. The overall accuracy, P_M and P_F rates vs. $P[H_1]$ are now presented for this sweep. Figure 4.11 illustrates the overall accuracy of the classifier for this second $P[H_1]$ sweep with w_{cell} set to 918 feet.

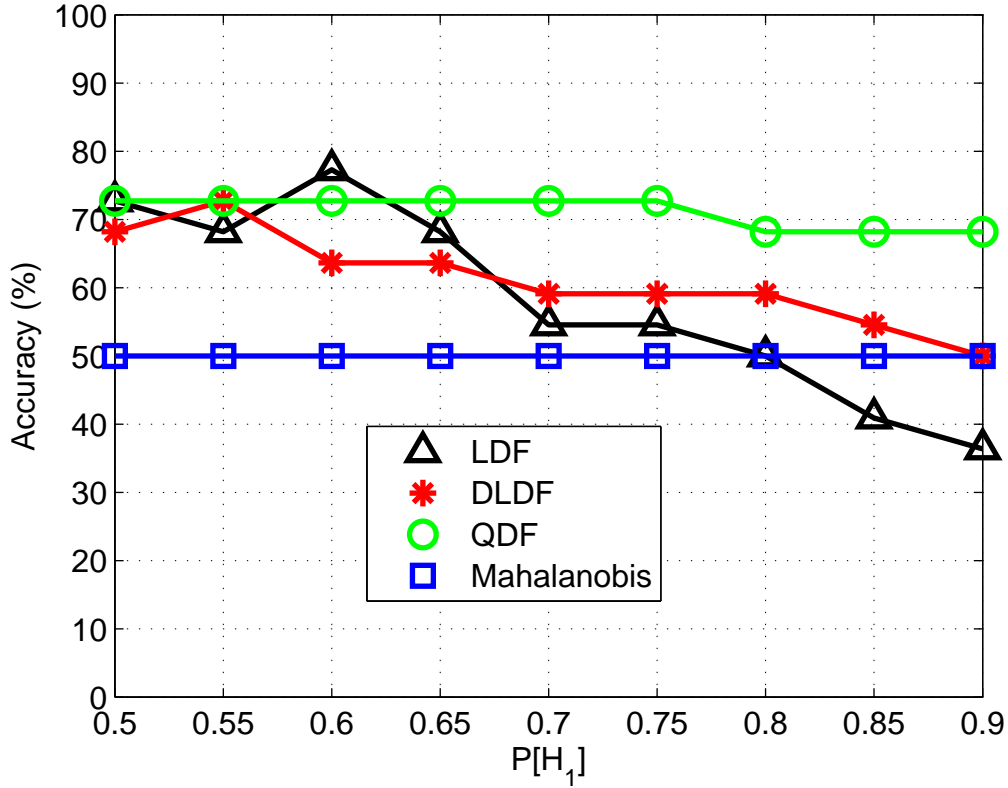


Figure 4.11: Classifier Accuracy for $P[H_1]$ Sweep: $w_{cell} = 918$ feet.

The QDF classification method again had the highest overall accuracy for this sweep. However, all four classification method's average accuracies were lower in this $P[H_1]$ sweep than the average accuracies in the previous sweep when w_{cell} was set to 270 feet. The average accuracy for the QDF method for this sweep was 71.21%. The QDF method had an accuracy of 72.73% at the start of the sweep. It remained at 72.73% until $P[H_1]$ was 0.75. At that point it decreased to and remained at 68.18% for the remainder of the sweep.

The standard deviation of the QDF classification method's accuracy for this sweep was 2.27%. This method's accuracy varied less for this sweep when compared to the accuracy for the grid resolution sweep that had a standard deviation of 8.58%.

The accuracy for the Mahalanobis method remained constant at 50.00%. The accuracy for the LDF classification method fluctuated at first and started descending to 36.36% when $P[H_1]$ was 0.6. Finally, the DLDF method's accuracy first increased slightly to 72.73% and then gradually decreased to 50.00% when $P[H_1]$ was 0.55.

Figure 4.12 presents the false negative error rate P_M of the classifier as a function of $P[H_1]$ for the four classification methods. w_{cell} was set to 918 feet.

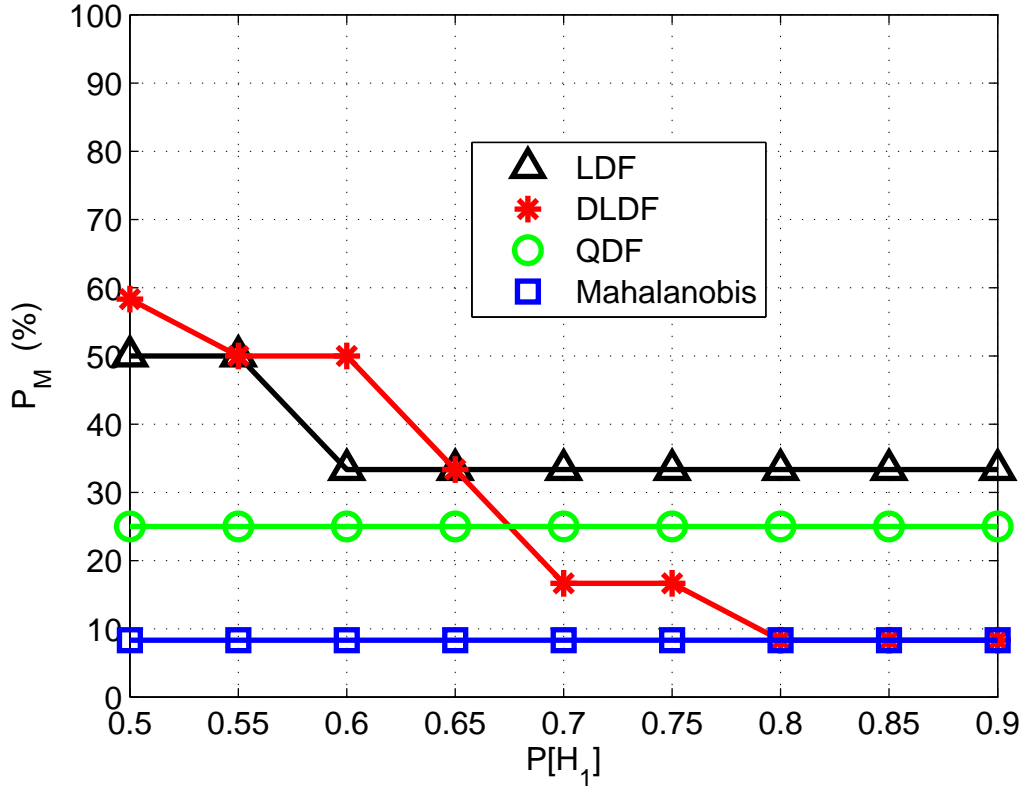


Figure 4.12: P_M Performance for $P[H_1]$ Sweep: $w_{cell} = 918$ feet.

The Mahalanobis classification method had the best average P_M rate: 8.33%. It remained at this value for the entire sweep. The QDF classification method had the second-best average P_M rate: 25.00%. It remained at this value for the entire sweep. The DLDF classification method had a P_M value of 60% at the beginning of the sweep and decreased to 10% as $P[H_1]$ increased. Finally, the LDF method had a P_M value of 50% when $P[H_1]$ was 0.5. As $P[H_1]$ increased, the P_M rate for this method decreased to 30%.

It was again expected that as $P[H_1]$ approached 0.9 during the sweep that the P_M rates for all classification methods would approach 0%. As $P[H_1]$ increased, the classifier became more sensitive to suspicious tracks and was more likely to classify any position track as suspicious. It was shown in this figure that the number of false negative errors

decreased for the DLDF and LDF classification methods as $P[H_1]$ increased. As stated earlier, the QDF and Mahalanobis classification methods both had P_M rates of 25% and 10% respectively during the entire sweep of $P[H_1]$.

Figure 4.13 illustrates the false positive error rates P_F of the four classification methods for the $P[H_1]$ sweep. w_{cell} was set to 918 feet.

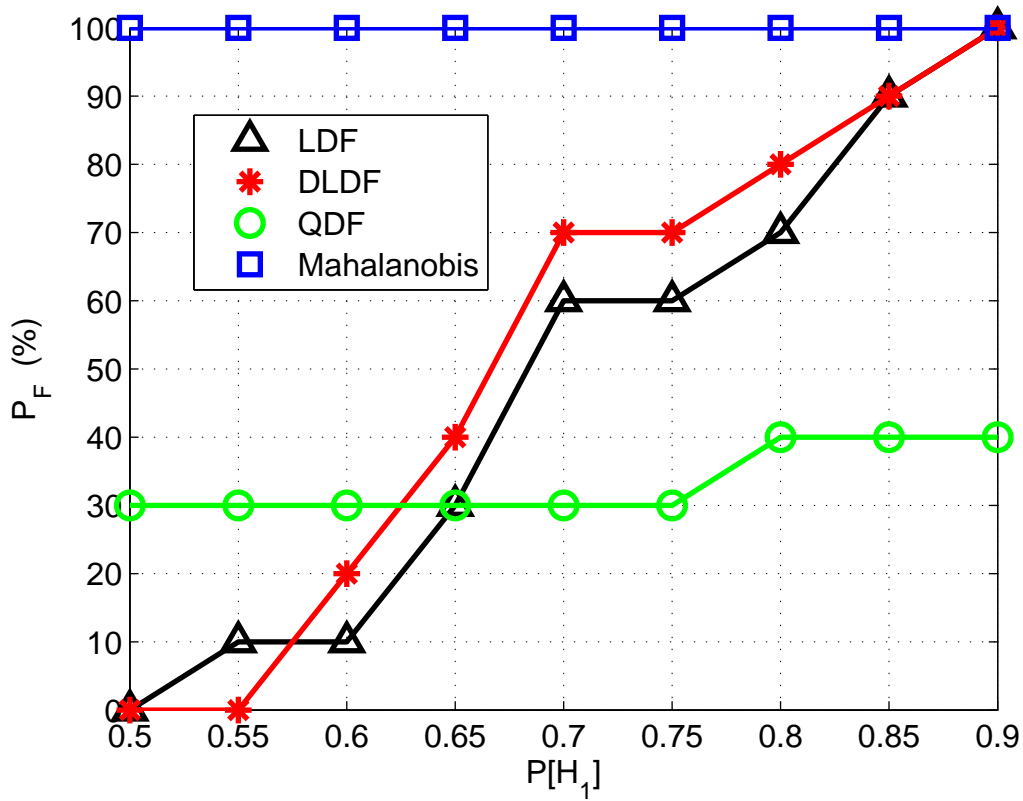


Figure 4.13: P_F Performance for $P[H_1]$ Sweep: $w_{cell} = 918$ feet.

The QDF classification method had the lowest average P_F value: 33.33%. This method had a P_F value of 30% at the beginning of the sweep. The P_F rate increased to 40% when $P[H_1]$ was 0.8 and remained there for the rest of the sweep. The ideal $P[H_1]$ range for this method was 0.5 to 0.75. When $P[H_1]$ was within this range, the QDF method had a false negative error rate of 25% (Figure 4.12), and a false positive error rate of 30%. 25% of suspicious position tracks in the database were incorrectly classified as non-suspicious, and 30% of non-suspicious tracks in the database were incorrectly classified as suspicious.

The LDF classification method had the next best average P_F rate: 47.78%. The P_F rate for this method increased steeply when $P[H_1]$ increased from 0.6 to 0.7. When $P[H_1]$ was 0.9, the LDF method had a 100% false alarm rate. The DLDF classification method

had an average P_F rate of 52.22%. The P_F rate for this method increased steeply when $P[H_1]$ increased from 0.55 to 0.7. When $P[H_1]$ was 0.9, the DLDF method had a 100% false alarm rate.

The Mahalanobis classification method had the worst average P_F rate: 100%. The Mahalanobis method remained at 100% for the entire sweep. It consistently classified all of the non-suspicious tracks in the database as suspicious during the sweep. The standard deviation of the P_F rates for the LDF, DLDF and QDF classification methods were greater than the standard deviation P_M rates for the $P[H_1]$ sweep. The Mahalanobis method had P_M and P_F standard deviations of 0% for this second $P[H_1]$ sweep.

It was expected that the P_F rates for all classification methods would increase as $P[H_1]$ approached 0.9 during the sweep. As $P[H_1]$ increased, the classifier became more sensitive to suspicious tracks and was more likely to classify any position track as suspicious. It was shown in Figure 4.13 that the number of false positive errors increased for all of the classification methods as $P[H_1]$ increased except the Mahalanobis method. The quantitative results for the this $P[H_1]$ sweep with w_{cell} set to 918 feet are summarized in Table 4.5.

| Statistics | LDF | DLDF | QDF | Mahalanobis |
|------------------------------------|-------|-------|-------|-------------|
| Average Accuracy (%) | 58.08 | 61.11 | 71.21 | 50.00 |
| Standard Deviation of Accuracy (%) | 14.33 | 6.86 | 2.27 | 0 |
| Average P_M (%) | 37.04 | 27.78 | 25.00 | 8.33 |
| P_M Standard Deviation (%) | 7.35 | 20.41 | 0 | 0 |
| Average P_F (%) | 47.78 | 52.22 | 33.33 | 100 |
| P_F Standard Deviation (%) | 36.67 | 38.33 | 5.00 | 0 |

Table 4.5: Classifier Statistics for $P[H_1]$ Sweep: $w_{cell} = 918$ feet.

Classifier statistics for the $P[H_1]$ parameter sweep of two different grid resolutions were presented and discussed in this subsection. w_{cell} was set to 270 feet for the first sweep and 918 feet for the second sweep. The next subsection presents and analyzes ROC curves generated using the P_F and P_M data from the $P[H_1]$ sweeps at these two grid resolutions.

4.2.3 ROC Curve Analysis

This subsection used P_F and P_M data to calculate best fit ROC curves for the four classification methods. First, $P[H_1]$ was swept from 0.0 to 1.0 in 0.05 increments in order to generate the largest number of unique P_F - P_M data pairs. This parameter sweep resulted in 21 pairs of P_F and P_M data. In some cases, the P_F and P_M data pairs had one or both values in the pair that were identical to one or both values in a different data pair. In chapter II, it was shown in Equation (2.16) that the P_D rate could be determined if the P_M rate was known.

$$P_D = 1 - P_M \quad (4.1)$$

P_D data was generated using this equation. The P_F and P_D data points were then plotted in the x-y coordinate format (P_F, P_D). This process was repeated for all four classification methods. The MATLAB® command `polyfit` was used to generate best-fit curves from the P_F and P_D data for each classification method. Through trial and error, the most appropriate polynomial degree was chosen for each classification method's best fit curve through the data points.

Figure 4.14 shows the P_F and P_D data points and corresponding best fit curves for each of the classification methods when w_{cell} was set to 270 feet.

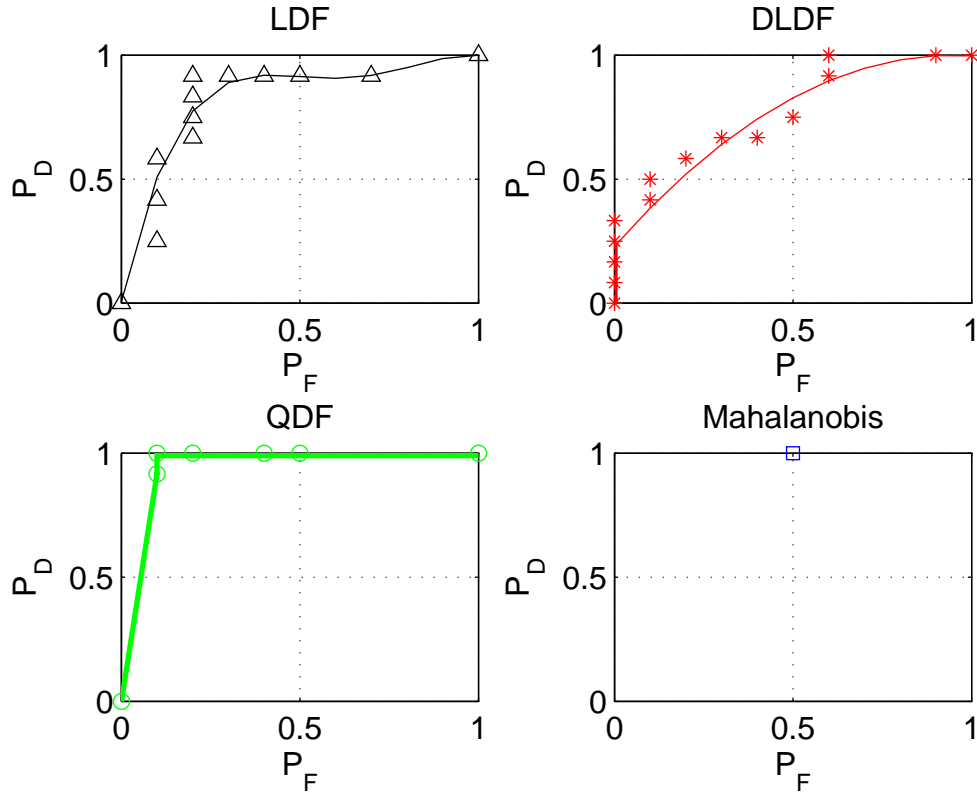


Figure 4.14: P_D vs. P_F Data Points and Best Fit ROC Curves: $w_{cell} = 270$ feet.

There were 13 unique P_F - P_D data points for the LDF classification method that were used to generate the ROC curve. A polynomial of degree four was the most appropriate for the P_F - P_D data points. The ROC curve for the LDF method was concave-down. There were 15 unique P_F - P_D data points for the DLDF classification method that were used to generate the ROC curve. A polynomial of degree two was the most appropriate for the P_F - P_D data points. The ROC curve for the DLDF method was also concave-down.

There were seven unique P_F - P_D data points for the QDF classification method that were used to generate the ROC curve. A polynomial of degree four was the most appropriate for the P_F - P_D data points. The ROC curve for the QDF method was concave-down. There was no curve generated for the Mahalanobis classification method because

this method operated at only one P_F - P_D data point. The data point for this grid resolution configuration was (0.5, 1).

Figure 4.15 shows the best fit curves for each of the classification methods with the random guess line displayed when w_{cell} was set to 270 feet.

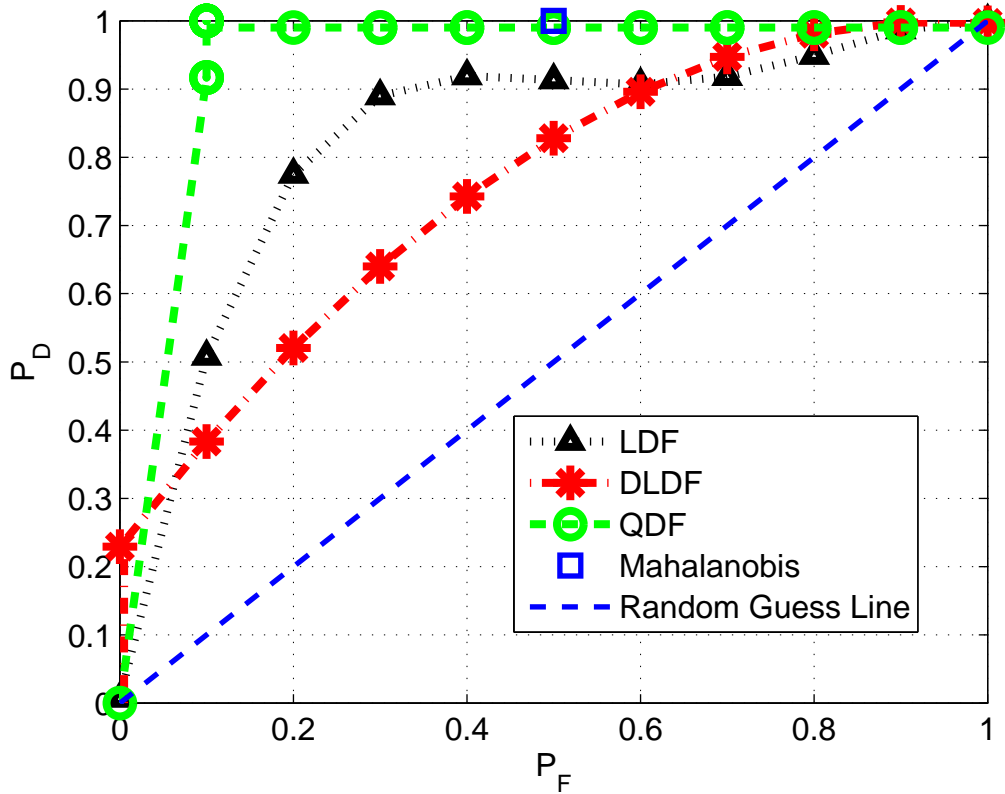


Figure 4.15: Best Fit ROC Curves: $w_{cell} = 270$ feet.

The three generated ROC curves and Mahalanobis point resided above the random guess line for the entire range of P_F . The QDF classification method had the most efficient ROC curve. The point on the curve where this classification method performed the best was at the P_F - P_D coordinates (0.1, 1). The method had a P_D of 100% and a P_F of only 10%. At this point on the curve, the QDF method classified every suspicious track as suspicious, and only 10% of non-suspicious tracks were classified as suspicious.

The Mahalanobis method operated at only one P_F - P_D data point. This point was at the P_F - P_D coordinates (0.5, 1). At this point, the classification method had a P_D of 100% and a P_F of 50%. The method classified every suspicious track as suspicious and 50% or

half of the non-suspicious tracks as suspicious. While this point was not as accurate as the (0.1, 1) point on the QDF ROC curve, it was still above the random guess line.

The LDF classification method's best fit ROC curve was not as ideal as the QDF method's curve but was still above the random guess lines for the entire range of P_F . The method achieved a P_D rate of about 0.77 when P_F was 0.2, and a P_D of about 0.92 when P_F was 0.40. The DLDF classification method's best fit ROC curve also was not as ideal as the QDF method's curve but was still above the random guess lines for all P_F values. The method achieved a P_D rate of about 0.75 when P_F was 0.4, and a P_D of 0.9 when P_F was 0.46.

Figure 4.16 shows the P_F and P_D data points and corresponding best fit curves for each of the classification methods when w_{cell} was set to 918 feet.

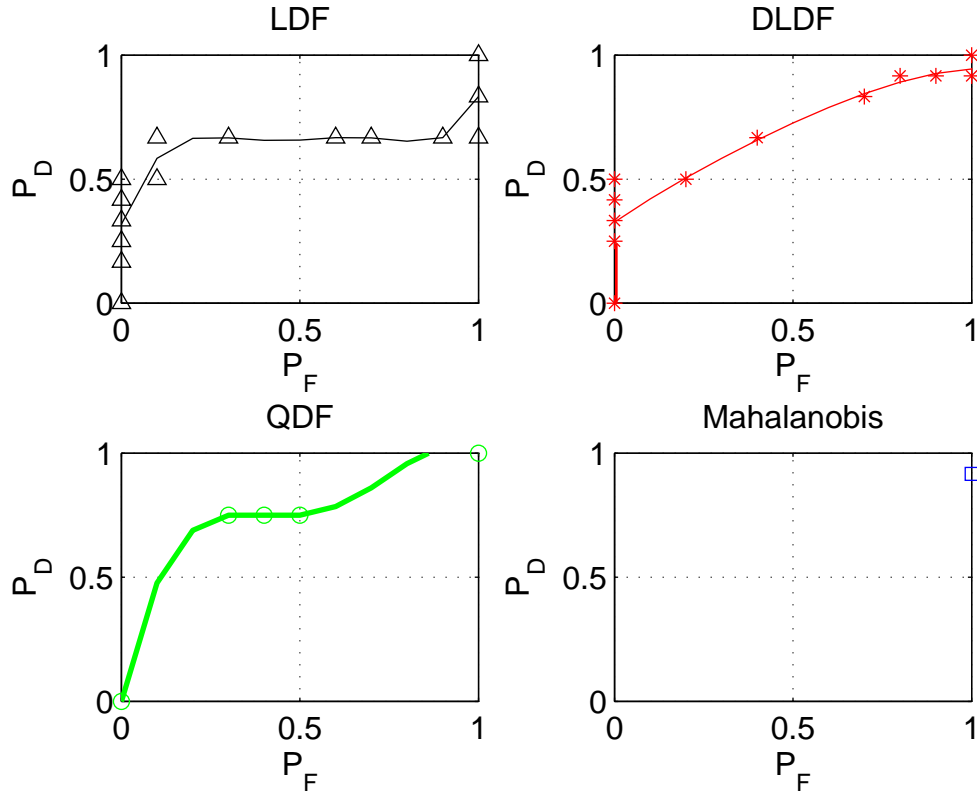


Figure 4.16: P_D vs. P_F Data Points and Best Fit ROC Curves: $w_{cell} = 918$ feet.

There were 15 unique P_F - P_D data points for the LDF classification method that were used to generate the ROC curve. A polynomial of degree six was the most appropriate for the P_F - P_D data points. The ROC curve for the LDF method was concave-down. There were 12 unique P_F - P_D data points for the DLDF classification method that were used to generate its ROC curve. A polynomial of degree four was the most appropriate for the P_F - P_D data points. The ROC curve for the DLDF method was also concave-down.

There were five unique P_F - P_D data points for the QDF classification method that were used to generate the ROC curve. A polynomial of degree four was the most appropriate for the P_F - P_D data points. The ROC curve for the QDF method was also concave-down.

es. There was again no curve generated for the Mahalanobis classification method because this method operated at only one P_F - P_D data point. The data point for this grid resolution configuration was (1, 0.92).

Figure 4.17 shows the best fit curves for each of the classification methods with the random guess line when w_{cell} was set to 918 feet.

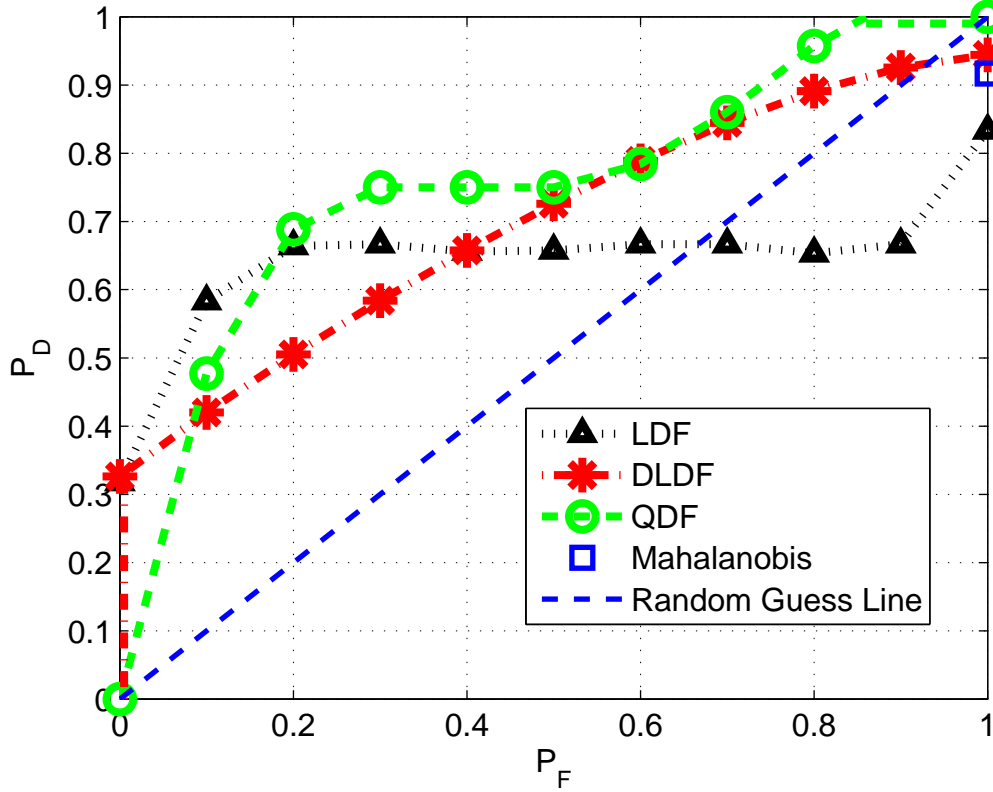


Figure 4.17: Best Fit ROC Curves: $w_{cell} = 918$ feet.

The best fit ROC curve for the QDF classification method at this grid configuration was the only curve that did not pass under the random guess line for the entire range of P_F . The best fit ROC curves for the LDF and DLDF classification methods eventually passed under the random guess line as P_F increased from 0 to 1. The Mahalanobis data point also resided under the random guess line.

The QDF classification method was not as ideal for this set of P_F - P_D data compared to the grid configuration when w_{cell} was set to 270 feet. When P_F was 0.2, P_D was about 0.70. When P_F was 0.4, P_D for the QDF method only increased to 0.75. It can be asserted that the most ideal point on the curve was (0.2, 0.70). This point on the curve was above the random guess line.

The DLDF classification method did not perform as well for this set of P_F - P_D data compared to when w_{cell} was set to 270 feet. When P_F was 0.2, P_D was about 0.5. When P_F was 0.6, P_D for this method was about 0.8. It can be asserted that the most ideal point on the curve was (0.4, 0.65). This point on the curve was still above the random guess line.

The LDF classification method also did not perform as well for this set of P_F - P_D data compared to when w_{cell} was set to 270 feet. The curve achieved a P_D value of about 0.65 relatively early, when P_F was 0.2. As P_F increased, P_D for this method remained about 0.65 as P_F approached 1. It passed under the random guess line when P_F was 0.66. It can be asserted that the most ideal point on the curve was (0.3, 0.66). This point on the curve was still above the random guess line.

Finally, the Mahalanobis method again operated at only one P_D vs. P_F point on the figure. This point was at the coordinates (1, 0.92) and was located under the random guess line. At this point, the classification method had a P_D of 92% and a P_F of 100%. The method incorrectly classified every non-suspicious track in the database as suspicious, and correctly classified 91.67 % or almost all suspicious tracks in the database as suspicious.

The QDF classification method had the most accurate best fit ROC curve for this grid resolution configuration. At the point where it was most efficient, about 70% of suspicious tracks were correctly classified as suspicious while only 20% of non-suspicious tracks were incorrectly classified as suspicious.

The results of the generated P_F and P_D data and best-fit ROC curves when w_{cell} was set to 270 and 918 feet were presented and analyzed in this subsection. P_F and P_M data were generated for each classification method from a $P[H_1]$ parameter sweep from 0 to 1. P_D data for each method was then calculated and the P_F - P_D data points were plotted. Polynomial functions that represented best fit ROC curves were fit to the data points using trial and error. The best fit ROC curve for each classification method for both grid resolution configurations were presented and analyzed.

4.3 Chapter Summary

This chapter presented the results that were attained in this research. First, four cross-sectional feature generation data plots were presented and discussed in section 4.1. Then, the classifier results and statistics were presented in section 4.2. This subsection first presented the classifier results for the grid resolution sweep. As emphasized in subsection 4.2.1, the QDF classification method outperformed the other three classification methods in overall accuracy. Then, two $P[H_1]$ sweeps were conducted for w_{cell} values of 270 and 918 feet. Finally, P_F - P_D data points were generated and calculated from a $P[H_1]$ sweep from 0 to 1 and best fit ROC curves were applied to the data in subsection 4.2.3.

This chapter communicated that the QDF classification method outperformed the other classification methods used in this research. This method performed the best when the grid cell width w_{cell} was set to 270 feet. At this grid resolution, the QDF accuracy was 95.45% when it classified the 22 position tracks one at a time. The average P_M rate during the $P[H_1]$ sweep for the QDF method at $w_{cell} = 270$ feet was 0% and the average P_F rate was 22.22%. The lowest P_M and P_F rates for the QDF method were 10% and 0%, respectively. The prior probabilities for the non-suspicious and suspicious classes ($P[H_0]$ and $P[H_1]$ respectively) were both set to 0.5 or 50% class for this system configuration.

Chapter V is the final chapter and presents a summary and conclusion of this research. Additionally, a future work section is included which outlines areas of future research that can be conducted.

V. Summary, Conclusions and Future Work

This chapter summarizes this thesis and the results produced by it. It also details significant conclusions drawn from the research and areas for future work.

5.1 Summary

The purpose of this research was to determine the feasibility of an RF emitter tracking and intent assessment as a means for enhanced physical security of a military installation. Chapter I introduced a brief background on this research area and included the problem statement, the scope and application of the research, the research objectives, equipment needed and the motivation for this thesis.

A more detailed background of source localization and pattern recognition concepts were provided in chapter II. Additionally, current research thrusts pertaining to employing pattern recognition techniques within a WSN to perform anomaly detection both in the physical and cyber domains were discussed. Chapter III communicated the methodologies used in this research. The chapter covered the standard operation of the Magellan® Mobile GPS unit, position track processing, feature generation from the position track database and classification of the position tracks.

The results of the intent assessment were presented chapter IV. First, the discriminant lines and curves for the four classification methods used in this research were plotted over cross-sectional generated feature data. These figures displayed data for two features at a time. The classifier statistics generated from two parameter sweeps were then presented. The two parameters were the grid cell spacing parameter (w_{cell}) and the suspicious class prior probability ($P[H_1]$). The classifier's accuracy as well as P_M and P_F errors were analyzed for both the grid resolution and $P[H_1]$ sweeps.

Finally, best fit ROC curves were generated using the P_F and P_M data from the $P[H_1]$ sweep. The P_M rates were converted into P_D rates and the data points were plotted in the x-y coordinate plane. Best fit ROC curves were generated to fit the P_F and P_D data for each of the four classification methods. The best fit ROC curves for each classification method were analyzed and compared.

5.2 Conclusions

This research has shown that it is possible to correctly classify position tracks as non-suspicious or suspicious using the feature data generated from them. In this research, data from five different features was generated for each position track. However, a position track can be classified as non-suspicious or suspicious using just one set of feature data. As the number of features used in the feature generation processes increased, the classifier accuracy also increased.

Chapter IV explained that the QDF classification method outperformed the other classification methods used in this research. This method performed the best when the grid cell width w_{cell} was set to 270 feet. At this grid resolution, the QDF accuracy was 95.45% when it classified the 22 position tracks one at a time. The average P_M rate during the $P[H_1]$ sweep for the QDF method at $w_{cell} = 270$ feet was 0% and the average P_F rate was 22.22%. The lowest P_M and P_F rates for the QDF method were 10% and 0%, respectively. The prior probabilities for the non-suspicious and suspicious classes ($P[H_0]$ and $P[H_1]$ respectively) were both set to 0.5 or 50% class for this system configuration.

This research confirms the feasibility and practicality of implementing an RF emitter tracking and intent assessment for a military installation with the design to improve physical security. The data and results that were produced by this research and communicated in chapter IV show that accurate feature data can be generated from position tracks and passed to a classifier using the QDF classification method to determine if the

behavior is suspicious or non-suspicious. The next section discusses areas for future work and research pertaining to this thesis.

5.3 Future Work

There are many areas that future work can be conducted in this research field. The following subsections communicate how further advancements can be realized through additional research.

5.3.1 Implementing Accurate Real-time Geolocation with TelosB Motes

A grid of wireless Memsic® TelosB sensors that geolocate an emitter in motion can replace the requirement for collecting position tracks with the GPS unit for the position track database. The base station would collect the RSS values from each sensor and pass a complete RSS data set in real-time to an estimation algorithm. One possible algorithm is the MLE. A position track could then be created of the x-y position estimates produced by the estimator. The feature generation and classification processes described in chapters II and III of this thesis could be applied in the same manner.

5.3.2 Tracking Multiple Emitters Simultaneously in a WSN

Another specific area that can be researched is the ability to simultaneously track multiple emitters in real-time. For this to be possible, the intent assessment system must include functionality to differentiate one emitter from another. The system would keep track of differences in communication protocol or signal structure between two or more transmitters. One example of a possible distinguisher for different RF devices is the Media Access Control (MAC) address.

The RSS values from the sensors would be separated into exclusive RSS data sets for each emitter in the network. A position estimator would input the data sets separately in order to produce device-specific x-y position estimates. The position tracks created

from geolocating multiple emitters would also be processed separately. Mutually exclusive feature data would be generated separately for each emitter's position track.

Each emitter in the WSN would be classified as suspicious or non-suspicious. If any one or combination of multiple emitters being tracked and monitored in the network were flagged as suspicious, the military installation security personnel would be notified.

5.3.3 Increasing the Size of the WSN

Increasing the size of the data collect area used for this research (defined by the overhead image presented in chapter III) would allow for additional landmark features to be employed in the classification process. Additionally, position tracks collected in the new area would have increased variability and diversity which would create broader data sets after the feature generation process.

5.3.4 Increasing the Size of the Position Track Database

Collecting more position tracks within the area defined by the overhead satellite image would increase the size of the position track database. Expanding the database would increase the training data set used by the classifier when it classifies an unknown track using the LOOCV method.

Increasing the number of position tracks in the database would also increase the resolution of the P_F and P_D data presented in chapter IV. In this research the resolution was only 0.1 because the database used in this research was comprised of only 10 non-suspicious and 12 suspicious tracks. As the number of position tracks in the database increases, the standard deviation of P_F would decrease, effectively increasing the confidence of the data.

5.3.5 Improving Feature Generation

The methods for generating feature data can be improved. Research can be conducted to improve the five existing feature generation algorithms discussed in chapter III, and also to create new features that can be used for classification.

A more sophisticated repetition algorithm can be implemented to generate more accurate repetition feature data. The algorithm would record the maximum euclidean distance that the position track reached before returning to a particular grid cell. Additionally, the length of time (in time-stamps) that the track remains from a particular grid cell can be recorded.

An extensive investigation can be performed to determine the complexity and efficiencies of the dwell and repetition feature algorithms. Specifically, the study can determine how the grid resolution affects the accuracy of the feature data generated by these two algorithms and the required processing-times on a micro-level.

An algorithm can be developed using the grid described in chapter III to determine the direction that an emitter is traveling. This can be accomplished by recording the grid cell that the emitter currently resides in, and then recording the adjacent or corner grid cell that the emitter moves to when it leaves the current cell. With this information, a direction can be determined that the emitter is traveling. This direction would be one of the four cardinal or four intermediate directions: north, east, west, south, north-east, etc.

The algorithm would then record the number of times that the emitter in motion changed its direction. Suspicious tracks generally possess more direction changes than non-suspicious tracks do. Therefore, this direction feature algorithm would assign a high score pertaining to suspicious activity for a position track that changed direction frequently. A low score would be assigned to a track that changed its direction minimally.

The velocity of a position track can be calculated using a Kalman filter. The filter will observe each position estimate over time and take the time derivative of the change in x-y position to determine the instantaneous velocity.

Additionally, more landmark features can be employed in the future. The buildings that comprise AFIT can be used as landmark features. Specifically, the exterior doors of the buildings can be set as non-suspicious in the landmark distance pixel map, and the rest

of the perimeters can be set as suspicious. In this manner, the landmark feature algorithm for the buildings will attribute a low score corresponding to non-suspicious activity for position tracks that come within close proximity to the doors, and subsequently a high score corresponding to suspicious activity when position tracks come within close proximity to a building's exterior wall or window.

Other landmark features that can be incorporated into the classifier are the WPAFB Area B perimeter fence and additional high-valued areas such as other buildings and power substations. With the addition of these new features, the accuracy of the classifier would improve.

5.3.6 Using Different Classification Methods

Different classification methods can be implemented in MATLAB® for the intent classifier. The results produced by these new methods can be incorporated with the results presented in chapter IV to create a more comprehensive view of the intent assessment performance. Radial Basis Functions (RBFs) and NNs can be implemented to perform position track classification. Different types of cross validation can be performed on the data collected in this research. The LOOCV method was the only method employed in this thesis. Two-fold cross validation is one type of cross validation that can be implemented.

5.4 Chapter Summary

This chapter summarized this thesis and the results produced by it. Significant conclusions drawn from the research and areas for future work were presented.

Appendix: Detailed Classifier Results

| Position Track # | Class |
|------------------|----------------|
| 1 | Suspicious |
| 2 | Suspicious |
| 3 | Suspicious |
| 4 | Non-Suspicious |
| 5 | Non-Suspicious |
| 6 | Non-Suspicious |
| 7 | Non-Suspicious |
| 8 | Suspicious |
| 9 | Suspicious |
| 10 | Suspicious |
| 11 | Non-Suspicious |
| 12 | Suspicious |
| 13 | Non-Suspicious |
| 14 | Non-Suspicious |
| 15 | Suspicious |
| 16 | Non-Suspicious |
| 17 | Suspicious |
| 18 | Non-Suspicious |
| 19 | Suspicious |
| 20 | Suspicious |

Table A.1: Classifier Database

| Position Track # | Class |
|------------------|----------------|
| 21 | Non-Suspicious |
| 22 | Suspicious |

Table A.2: Classifier Database

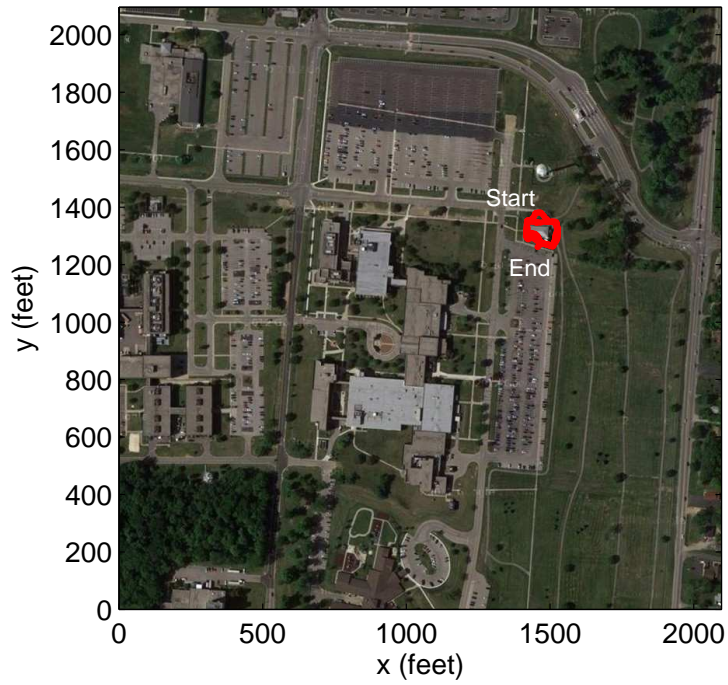


Figure A.1: Track 1 on Overhead Imagery.

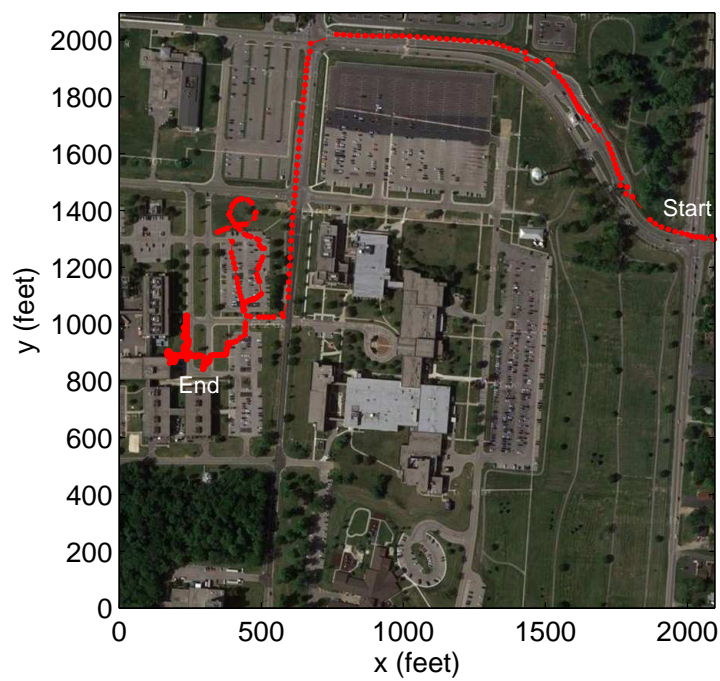


Figure A.2: Track 2 on Overhead Imagery.

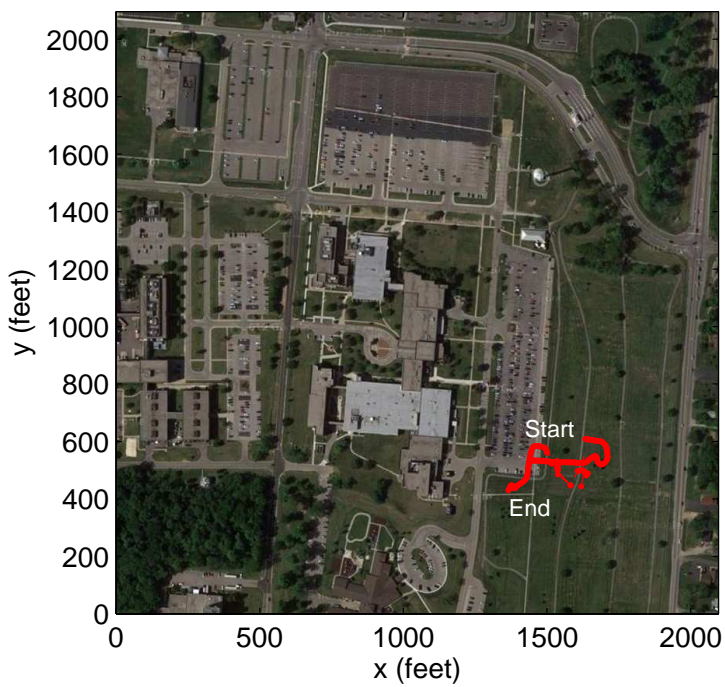


Figure A.3: Track 3 on Overhead Imagery.

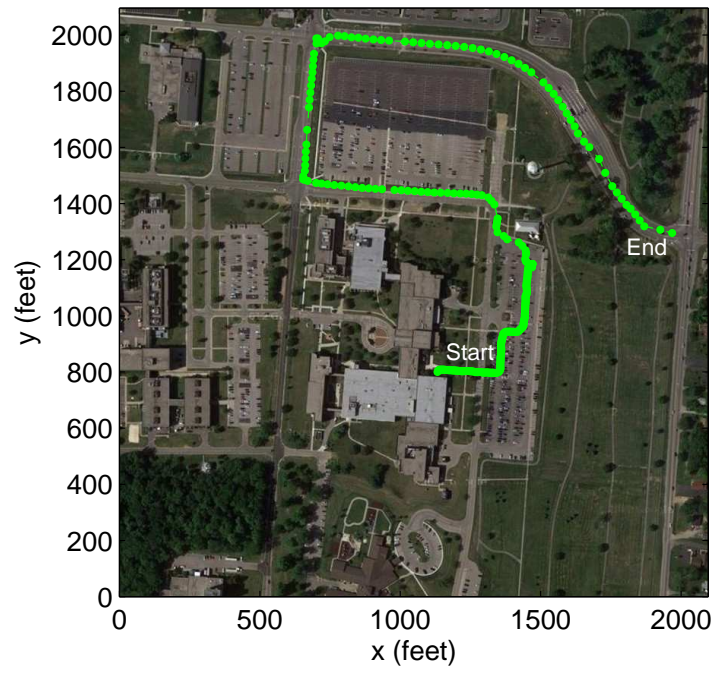


Figure A.4: Track 4 on Overhead Imagery.

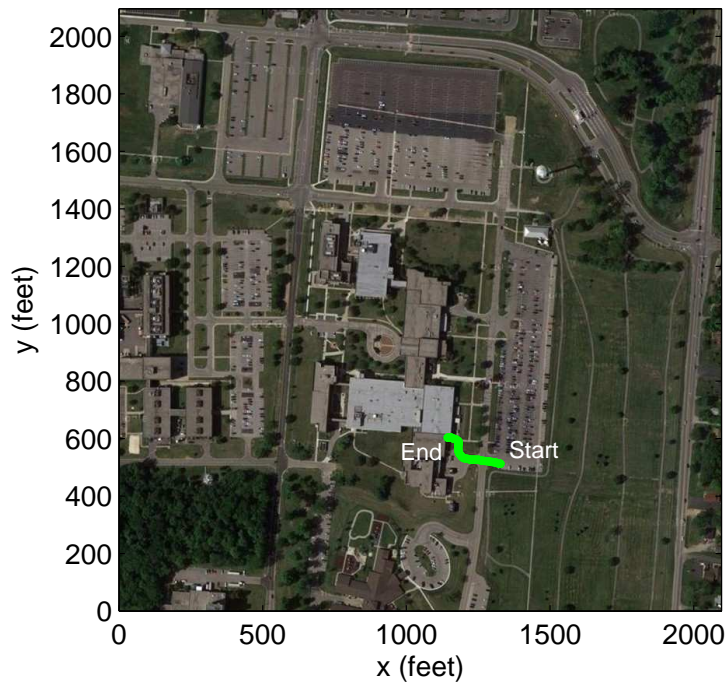


Figure A.5: Track 5 on Overhead Imagery.

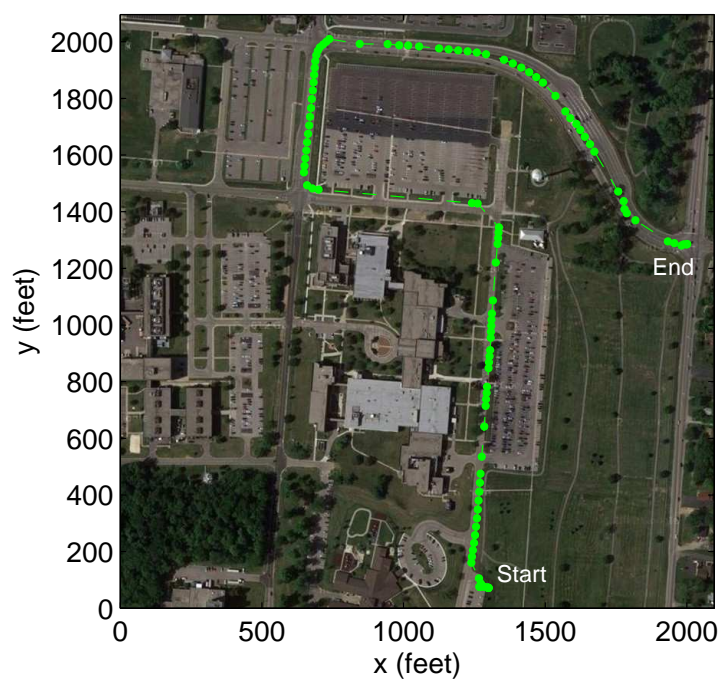


Figure A.6: Track 6 on Overhead Imagery.

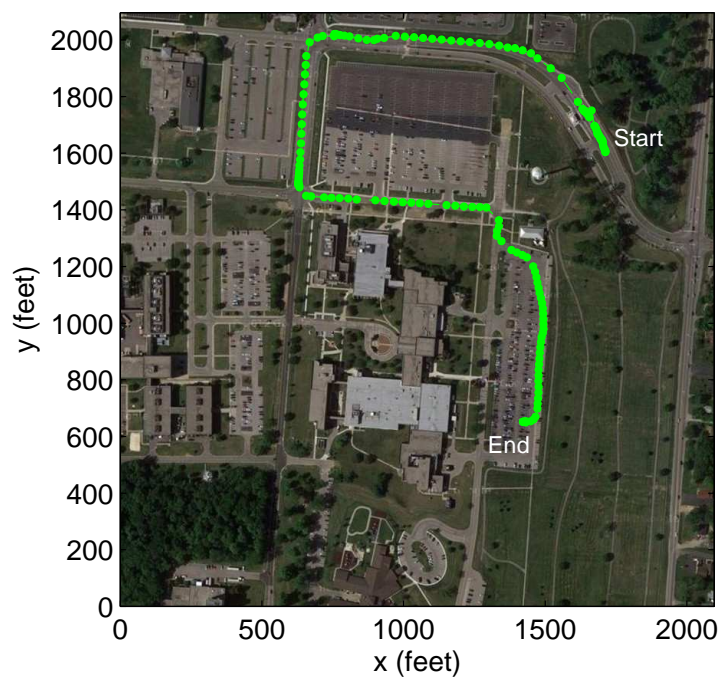


Figure A.7: Track 7 on Overhead Imagery.

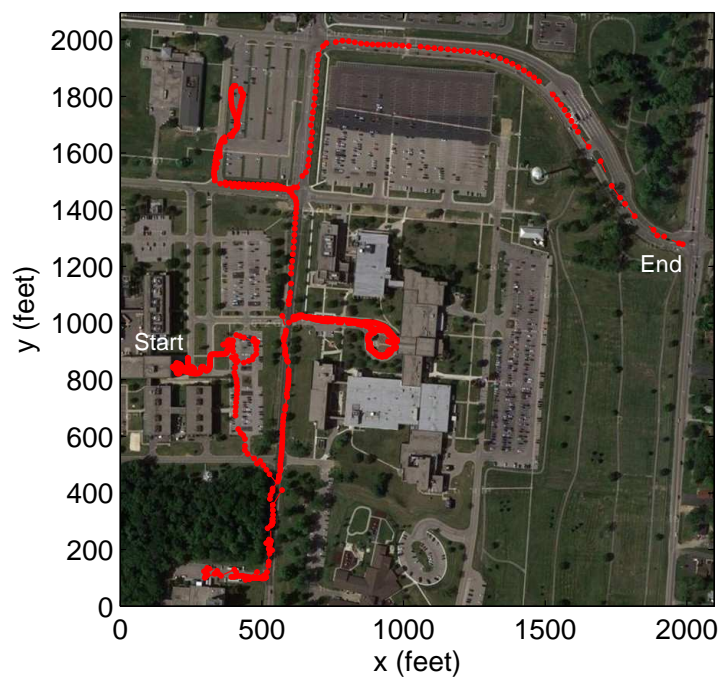


Figure A.8: Track 8 on Overhead Imagery.

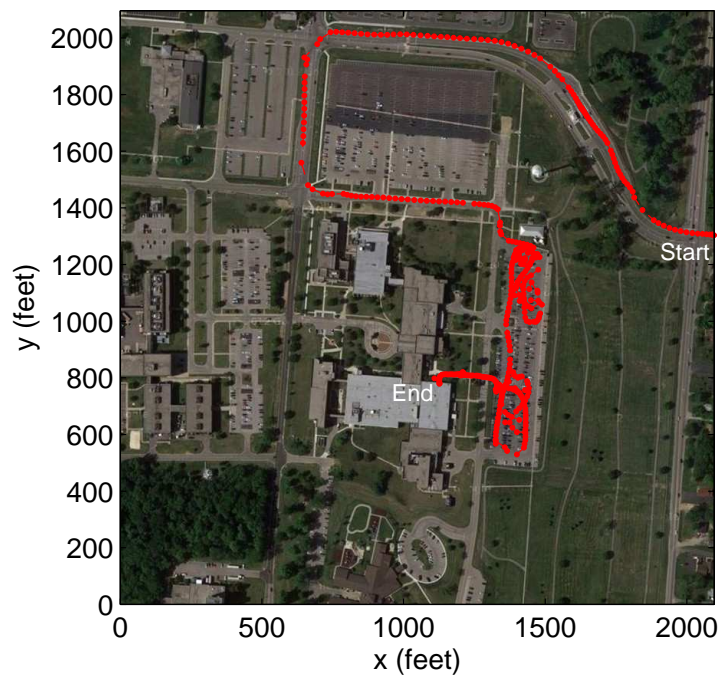


Figure A.9: Track 9 on Overhead Imagery.

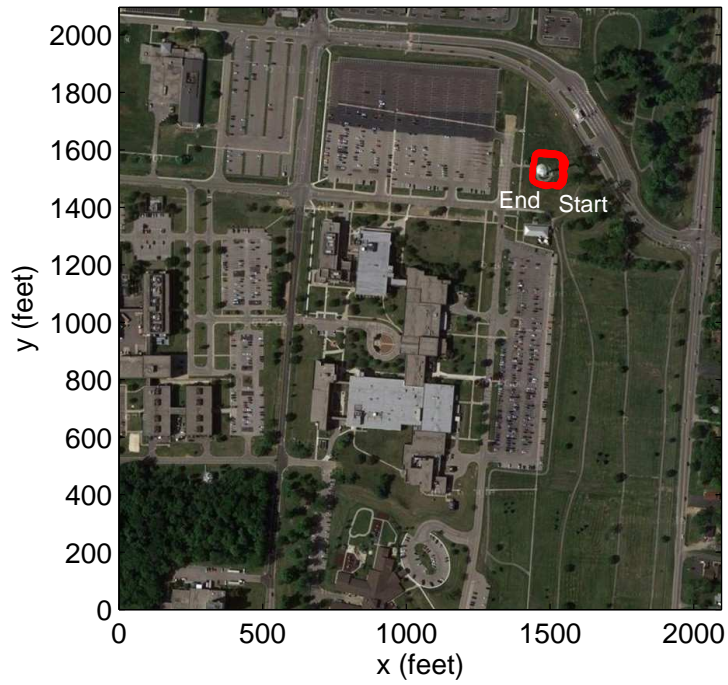


Figure A.10: Track 10 on Overhead Imagery.

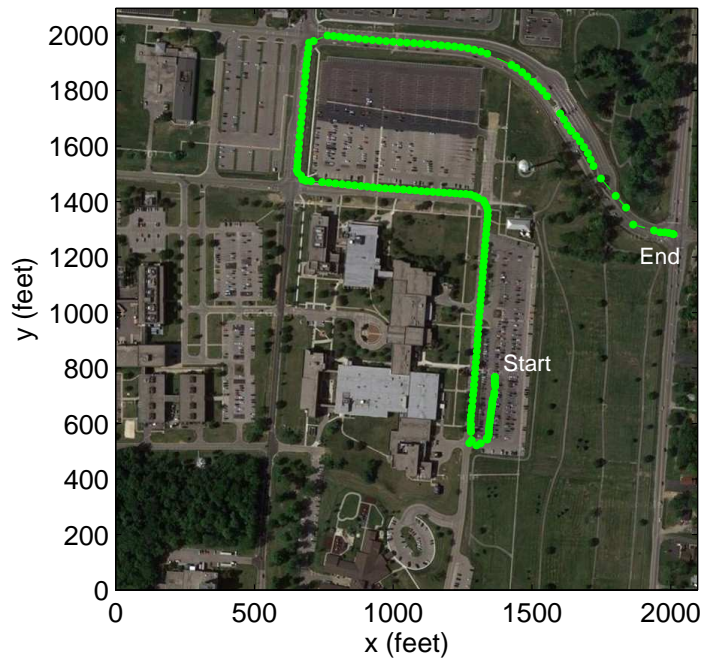


Figure A.11: Track 11 on Overhead Imagery.

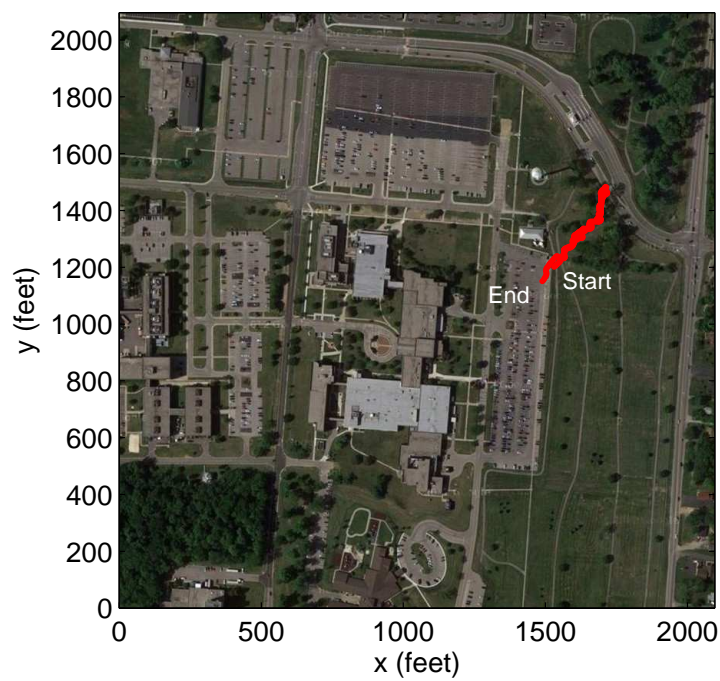


Figure A.12: Track 12 on Overhead Imagery.

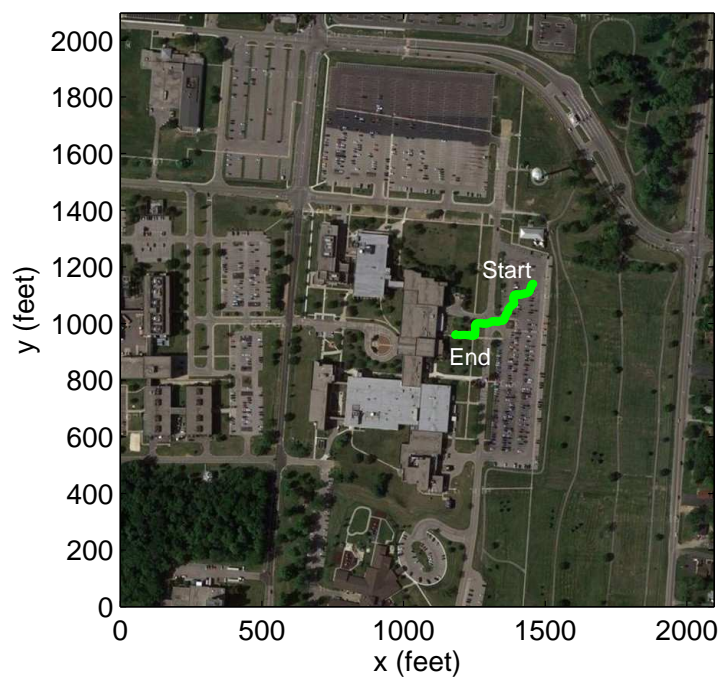


Figure A.13: Track 13 on Overhead Imagery.

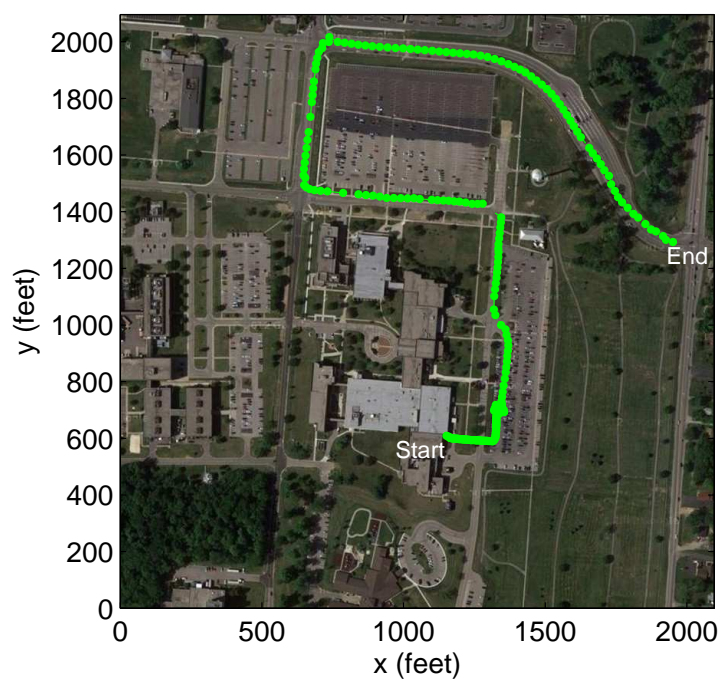


Figure A.14: Track 14 on Overhead Imagery.

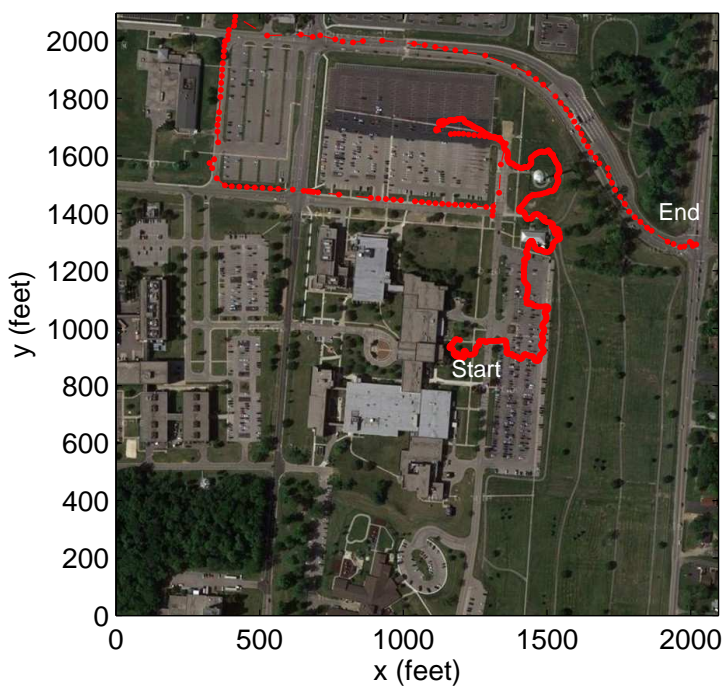


Figure A.15: Track 15 on Overhead Imagery.

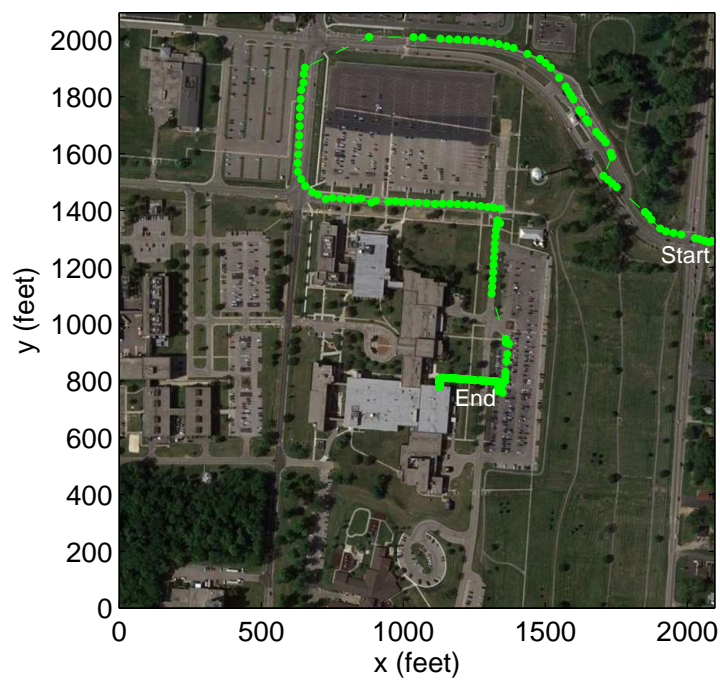


Figure A.16: Track 16 on Overhead Imagery.

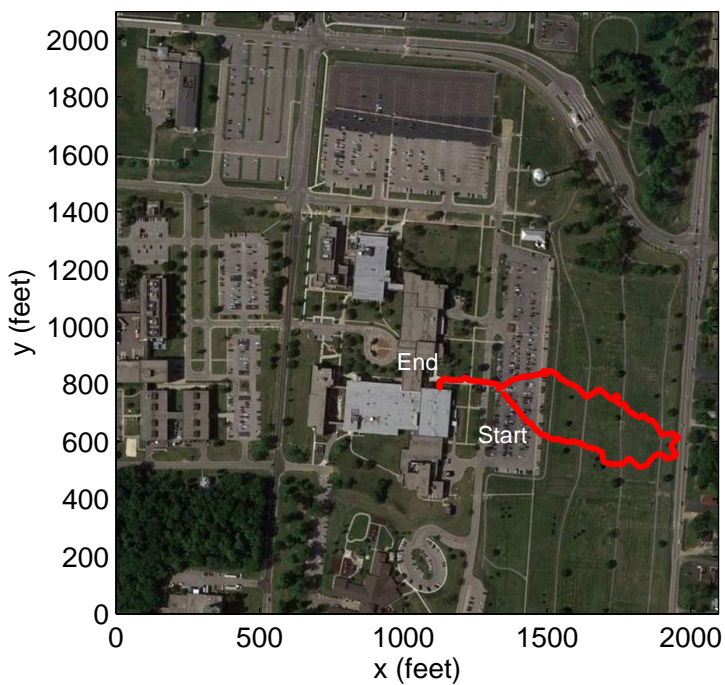


Figure A.17: Track 17 on Overhead Imagery.

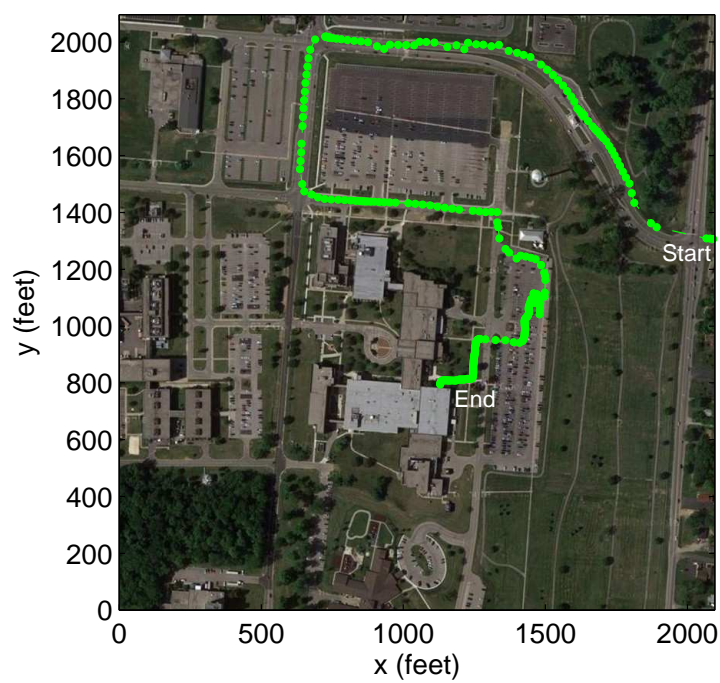


Figure A.18: Track 18 on Overhead Imagery.

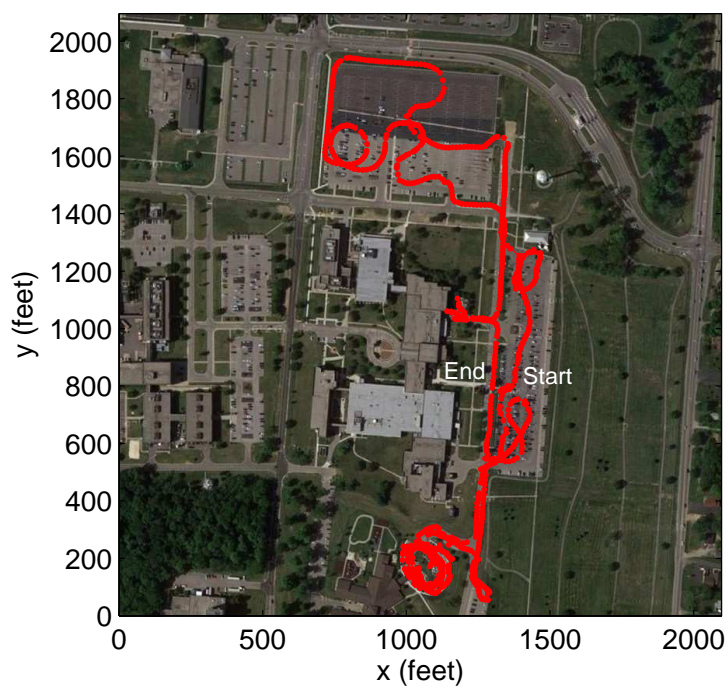


Figure A.19: Track 19 on Overhead Imagery.

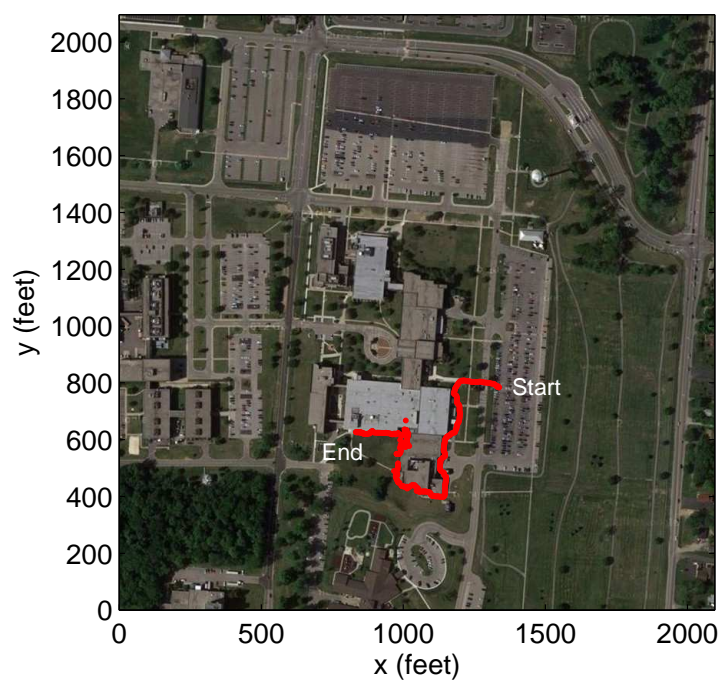


Figure A.20: Track 20 on Overhead Imagery.

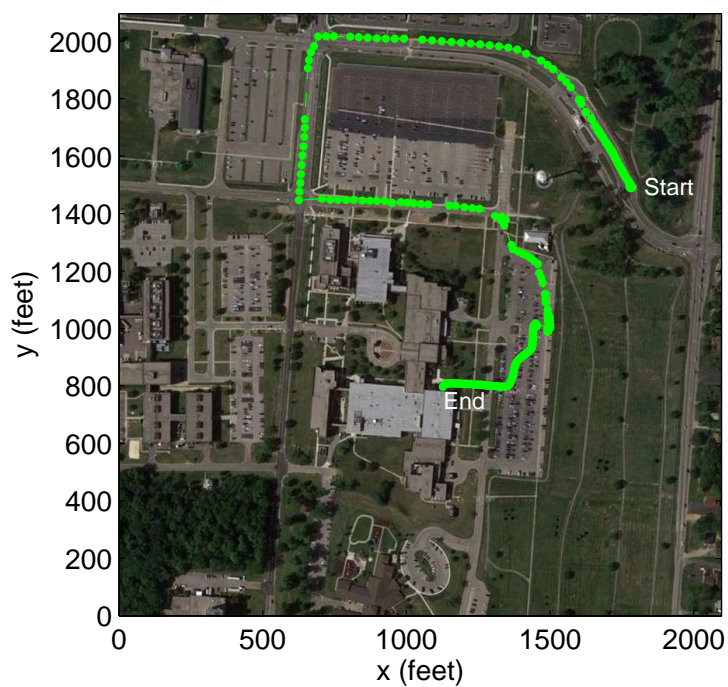


Figure A.21: Track 21 on Overhead Imagery.

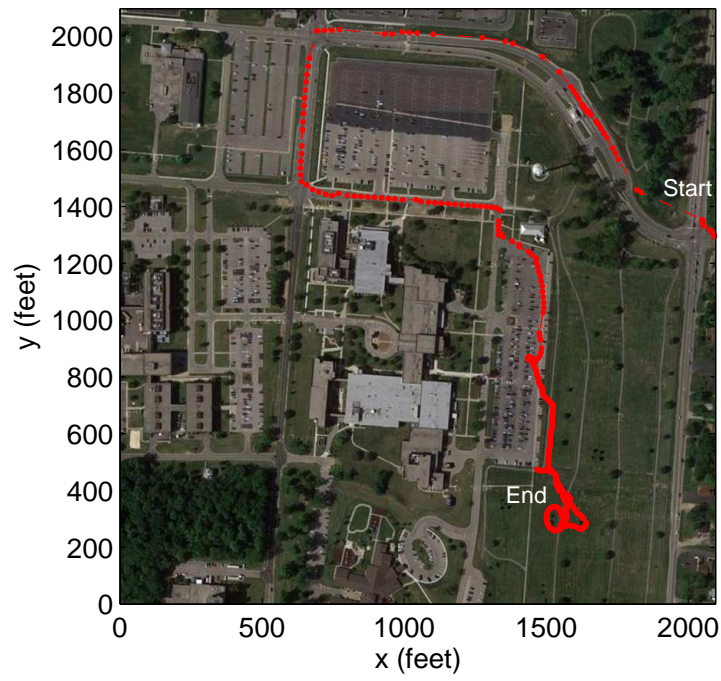


Figure A.22: Track 22 on Overhead Imagery.

Bibliography

1. M. S. Butler, "Low Cost, Low Complexity Sensor Design for Non-Cooperative Geolocation via Received Signal Strength," Master's thesis, Air Force Institute of Technology, March 2012.
2. F. Lardinois, "Number of Location-Aware Apps Keeps Growing Rapidly - But Very Few are Cross-Platform," February 2010. [Online]. Available: http://readwrite.com/2010/02/05/number_of_location-aware_apps_keeps_growing_-_but.
3. L. Wang and Q. Xu, "Gps-Free Localization Algorithm for Wireless Sensor Networks," *Sensors*, vol. 10, no. 6, pp. 5899–5926, 2010.
4. Z. A. Baig, M. Baqer, and A. I. Khan, "A Pattern Recognition Scheme for Distributed Denial of Service (DDoS) Attacks in Wireless Sensor Networks," *International Conference on Pattern Recognition*, vol. 3, pp. 1050–1054, 2006.
5. Z. A. Baig, "Pattern Recognition for Detecting Distributed Node Exhaustion Attacks in Wireless Sensor Networks," *Computer Communications*, vol. 34, no. 3, pp. 468–484, 2011.
6. R. K. Martin and R. W. Thomas, "RF Anomaly Detection for Intent Assessment," submitted to the Office of Naval Research, 2009.
7. R. K. Martin, "Emitter Geolocation via Signal Strength," 2012.
8. W. Tidd, R. Weber, Y. Huang, and Y. Zhao, "A RF Source Localization and Tracking System," in *Military Communications Conference, 2010 - MILCOM 2010*, November 2010, pp. 858–863.
9. (2008, May) Wi-Fi Location-Based Services (4.1 Design Guide). Cisco. [Online]. Available: <http://www.cisco.com/en/US/docs>.
10. T. Rappaport, J. H. Reed, and B. D. Woerner, "Position Location Using Wireless Communications on Highways of the Future," *IEEE Communications Magazine*, vol. 34, no. 10, pp. 33–41, October 1996.
11. S. M. Kay, *Fundamentals of Statistical Signal Processing Estimation Theory*, ser. 18th Printing, A. V. Oppenheim, Ed. Prentice Hall PTR, 2010.
12. B. J. Kuhar, "Emitter Geolocation via Signal Strength," 2012.
13. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, 2001.

14. E. Bullmore, J. Suckling, S. Overmeyer, S. Rabe-Hesketh, E. Taylor, and M. Brammer, "Global, Voxel and Cluster Tests, by Theory and Permutation, for a Difference Between Two Groups of Structural MR Images of the Brain," *IEEE Transactions on Medical Imaging*, vol. 18, no. 1, pp. 32–42, January 1999.
15. A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A Review of Anomaly Detection in Automated Surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1257–1272, November 2012.
16. S. Su, X. Duan, X. Zeng, W. Chan, and K. Li, "Context Information-Based Cyber Security Defense of Protection System," in *Power Engineering Society General Meeting, 2007. IEEE*, June 2007, p. 1.

| | | | | | | |
|--|----------------------|--|---|---------------------------------------|--|--|
| REPORT DOCUMENTATION PAGE | | | | | Form Approved OMB No. 0704-0188 | |
| The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS. | | | | | | |
| 1. REPORT DATE (DD-MM-YYYY) 21-03-2013 | | 2. REPORT TYPE Master's Thesis | | | 3. DATES COVERED (From — To) Oct 2011–Mar 2013 | |
| 4. TITLE AND SUBTITLE RF Emitter Tracking and Intent Assessment | | | | 5a. CONTRACT NUMBER | | |
| | | | | 5b. GRANT NUMBER | | |
| | | | | 5c. PROGRAM ELEMENT NUMBER | | |
| 6. AUTHOR(S) Kuhar, Benjamin J., Captain, USAF | | | | 5d. PROJECT NUMBER | | |
| | | | | 5e. TASK NUMBER | | |
| | | | | 5f. WORK UNIT NUMBER | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765 | | | | | 8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENG-13-M-29 | |
| 9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Attn: AFRL RYWE (Dr. Vasu Chakravarthy) 2241 Avionics Circle WPAFB, OH 45433 (937) 785-5579 ext 4245 vasu.chakravarthy@wpafb.af.mil | | | | | 10. SPONSOR/MONITOR'S ACRONYM(S) | |
| | | | | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) | |
| 12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED | | | | | | |
| 13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States. | | | | | | |
| 14. ABSTRACT Current research in employing pattern recognition techniques in a wireless sensor network (WSN) to detect anomalous or suspicious behavior is limited. The purpose of this research was to determine the feasibility of an accurate tracking and intent assessment system of unknown or foreign radio frequency (RF) emitters in close proximity to and within military installations as a method for physical security. 22 position tracks were collected using a hand-held Global Positioning System (GPS) unit and a training data set from five different features was generated for each position track. Each collected position track was individually classified as suspicious or non-suspicious by the leave-one-out-cross-validation (LOOCV) method using four different classification methods. The four classification methods used in this research were the linear discriminant function (LDF), the diagonal linear discriminant function (DLDF), the quadratic discriminant function (QDF) and the Mahalanobis distance method. The accuracies and false positive/negative error rates of the four classification methods were compared for different assessment system configurations. Additionally, best fit receiver operating characteristic (ROC) curves were generated for each classification method and discussed. The QDF classification method out-performed the other three classification methods. This classification method achieved an accuracy of 95% when it classified the 22 position tracks one at a time. The lowest false positive and false negative rates were 10% and 0%, respectively. The prior probabilities for the non-suspicious and suspicious classes were both set to 50% class for this configuration. | | | | | | |
| 15. SUBJECT TERMS Radio Frequency, Tracking and Intent Assessment, Geolocation, Received Signal Strength, Anomaly Detection, Pattern Recognition, Classification | | | | | | |
| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT UU | 18. NUMBER OF PAGES 132 | 19a. NAME OF RESPONSIBLE PERSON Dr. Richard K. Martin (ENG) | |
| a. REPORT U | b. ABSTRACT U | c. THIS PAGE U | | | 19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x4625 Richard.Martin@afit.edu | |